1993

A study of continuous time adaptive filters

James M. Little
Lehigh University

AUTHOR:

Little, James M.

TITLE:

A Study of Continuous Time Adaptive Filters

DATE: May 30, 1993

A Study of Continuous Time Adaptive Filters

by

James M. Little

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the degree of

Master of Science in Electrical Engineering

Lehigh University

1993

This thesis is accepted and approved in partial fulfillment of the requirments for the degree of Masters of Science in Electrical Engineering.

5/12/93

Date

Dr. Douglas Frey
Advisor in Charge

Dr. Alastair McAulay
CSEE Department Chairperson

Acknowledgements

I wish to thank my advisor Dr. Douglas Frey for his interest and assistance in helping me explore this topic. Thanks are also due to Jonathan Fischer at AT&T Bell Laboratories for his proofreading and encouragement. I must also extend thanks to my management at AT&T Bell Laboratories for providing me the opportunity to attend Lehigh. Finally, I would like to thank my wife and partner Beata, and my children Ian and Daniel, for their patience and understanding.

Table of Contents

Abstract 1

1.0 Introduction 2

2.0 Derivation of the Adaptive Algorithms 5

2.1 Weiner Hopf Equation 5
2.2 Continuous Time Filter Optimal Solution 7
2.3 Steepest Descent Method 10
2.4 Adaptation Based on Least Mean Squares Estimate of The Error Gradient 11
2.5 Convergence and Stability Issues 15
2.6 Equation Error Adaptation 17
2.7 Variations of the Algorithm 20
2.8 Summary 20

3.0 Circuit Models for Adaptive Biquad 22

3.1 State Variable Filter With Adaptive Zeros 22
3.2 State Variable Filter With Adaptive Poles and Zeros 26
3.3 Zero Adapting Biquad 28
3.4 Pole-Zero Adaptive Filter 29
3.5 Equation Error Adaptive Filter 29
3.6 System Transfer Function 30
3.7 DC Sensitivity 31
3.8 Stability Issues 32
3.9 Summary 34

4.0 Simulation Methodology 35

4.1 Discrete Time Filter Models 35
4.2 Mapping the Filter 39
4.3 Input Signal Generation 40

5.0 Experimental Results 42

5.1 Continuous Time Filter With Adaptive Zeros 42
5.2 Output Error Experiments 70
5.3 Equation Error Adaptation Experiments 85

6.0 Conclusions 94

List of Tables

Table 1	Gain Equalizer Parameters	43
Table 2	Amplitude Equalizer Model Gradient Filters	46
Table 3	ABQ5B Equalizer Behavior	46
Table 4	Model Matching Stability	49
Table 5	Model Matching Convergence with Non-Zero Mean	53
Table 6	Multi-Tone Autocorrelation Matrix	58
Table 7	Autocorrelation Matrix for Gaussian Noise	58
Table 8	Natural Modes of Error Impulse Response	59
Table 9	Double Tone Convergence	64
Table 10	Output Error Experiment Filter Parameters	70
Table 11	Equation Error Experimental Data	85
Table 12	Equation Error Experimental Results	86

List of Figures

Figure 1 System Identification Problem 5
Figure 2 General Adaptive Filter 8
Figure 3 Equation Error Formulation 18
Figure 4 State Variable Adaptive Biquad 23
Figure 5 Error Signal Processing 24
Figure 6 LMS Adaptation Circuits and Gradient Filter 25
Figure 7 Numerator Gradient Filter 27
Figure 8 Denominator Term LMS Processing 28
Figure 9 Adaptive Zero Only Biquad 28
Figure 10 Pole- Zero Adaptive Filter 29
Figure 11 Equation Error Filter 30
Figure 12 Stability Triangle 32
Figure 13 Second Order IIR Filter 36
Figure 14 Second Order Biquadratic Adaptive Filter Model 37
Figure 15 Second Order Biquadratic Adaptive Filter Model 38
Figure 16 Input Noise Source Characteristics 41
Figure 17 Prototype Filter Transfer Function 44
Figure 18 Input and Desired Signals 45
Figure 19 Coefficient Trajectories Zero Only Adaptation 47
Figure 20 Learning Curves Zero Only Adaptation 48
Figure 21 Coefficients Trajectories with Small LMS gain 51
Figure 22 Multi-tone Input and Desired Signals 55
Figure 23 Multi-tone Coefficient Trajectory 56
Figure 24 Zero Movements -- Multitone Input Signal 57
Figure 25 Error Step Response LMS gain = 0.03125 60
Figure 26 Error Step Response LMS gain = 0.0625 61
Figure 27 Signal to Noise Experiments 62
Figure 28 Learning Curve for Various S/N Rations 63
Figure 29 Learning Curve for Two Tone Input 65
Figure 30 Interference Cancelling 66
Figure 31 Interference Cancelling Experiment Results 68
Figure 32 Interference Cancelling Experiment Transfer Function 69
Figure 33 Output Error Experiment Numerator Coefficient 71
Figure 34 Output Error Experiment Denominator Coefficient 72
Figure 35 Output Error Experiment Pole Movement 73
Figure 36 Output Error Experiment Zero Movement 74
Figure 37 Output Error Experiment Learning Curve 75
Figure 38 Output Error Formulation Transfer Functions 76

Figure 39 Pole Movement for Improper Initialization 82
Figure 40 Learning Curve for Improper Initialization..... 83
Figure 41 Pole Movement with Multi-tone Input..... 84
Figure 40 Learning Curves for Various S/N Ratios..... 85
Figure 41 Pole Movement for Equation Error Experiment #1 88
Figure 42 Pole Movement for Equation Error Experiment #2 89
Figure 43 Zeros Movement for Equation Error Experiment #1 90
Figure 44 Zeros Movement for Equation Error Experiment #2 91
Figure 45 Learning Curve for Equation Error Experiment #1 92
Figure 46 Learning Curve for Equation Error Experiment #2 93

Abstract

This thesis presents three methods for designing continuous time adaptive filters. These methods are based on applying the Least Mean Squares (LMS) algorithm to a second order state variable biquadratic filter. It is shown that each state of the filter is the gradient of the filter output with respect to the filter coefficient for that state. These gradients can be used with the LMS estimate of the error of the filter to generate update equations for both the numerator and denominator coefficients of the filter.These equations can be used to adapt the filter towards the optimal solution.

The first method presented has adaptive zeros and fixed poles and is shown to be robust and to have many similarities to standard adaptive FIR filters. The second method presented adapts the poles and zeros of the filter directly in the output error configuration. The third method presented adapts the poles and zeros indirectly in the equation error configuration. It is shown that proper initialization of the filter poles is important in both the second and third methods and that the best results are obtained when the poles are initialized as real poles with the locus of the desired final poles between them. Simulation results are presented showing the operation of each type of adaptive filter for a system identification problem. All three second order adaptive filter types were found to be able to identify an unknown second order system. Finally some directions for further work are presented.

1.0 Introduction

Since their introduction by Lucky (1966) and Widrow (1975) adaptive electronic filters have become a key part of many modern electronic systems. Before Lucky introduced the automatic transversal equalizer in 1966 [19] the problems of equalizing telephone networks to remove or reduce interference caused by dispersion in the transmission media had been analyzed by Bode [3] and others and a wide variety of adjustable solutions had been tried in an effort to synthesize a matched filter receiver. Lucky was the first to apply automatic adjustment of the tap gain(s) of a transversal filter to the problem. His analysis and solution of the problem is one of the classic works on adaptive equalizers. The algorithm he designed minimized the distortion in the channel and can be considered to be in the family of zero-forcing equalizers. Luckey's filter found wide application as an equalizer in central telephone offices. It was not until Widrow introduced the adaptive linear combiner as a noise cancelling system in 1975 [35] that adaptive filter technology began to become widely accepted. What Widrow did in his classic papers in 1975 and 1976 was to introduce the discrete time FIR filter with adjustable tap weights as a means of synthesizing the optimal Weiner solution to a given problem. In addition his extensive analysis of the convergence and transient behavior of the filter [36]showed the robust nature of the design. It wasn't long before the adaptive linear combiner and the least mean squares (LMS)algorithm were being used for a wide variety of applications including echo cancellers, equalizers, control systems and speech processors. The lack of high quality, high speed A/D converters caused many of these early systems to be implemented as analog systems either in the continuous time domain, with tapped delay lines forming FIR filters [33] or, in the discrete time domain with CMOS and CCD technology[38]. With the rapid expansion and acceptance of digital electronics and with the advent of the microprocessor and later the digital signal processor, the bulk of adaptive filter applications began to be implemented as purely digital systems.

While the flexibility of digital systems with stored programs is to be admired there are many applications and potential applications of adaptive filters for which the size, cost and power associated with such systems in prohibitive. Examples of such applications are hearing aids and modems for lap-top and palm-top computers. In addition, the burgeoning data communications market demands higher and higher transmission rates on the cheapest twisted pair wire creating a wide demand for low cost, low power, high speed equalizers and echo cancellers[24]. These applications are still beyond the speed of conventional ADC's and DSP chips and may be best served by analog systems.

Analog adaptive filters may be approached in two ways. First, as previously mentioned it is possible to build analog FIR filters either from continuous time delay lines or a switched capacitor circuits with or without CCD delay lines. Both of these methods have advantages and disadvantages. However because of the widespread acceptance of CMOS as the technology for single chip integrated systems it seems that the switched capacitor implementation of discrete time analog FIR filters is the preferred method of building adaptive analog FIR filters particularly at lower frequencies. Because switched capacitor systems are sampled data systems there is a tremendous burden upon the electronics at higher data rates. Present day technology limits the bandwidth of analog CMOS circuits to less than 100MHz. Since current technology limits switched capacitor FIR filters to bandwidths less than 10 - 20 MHz there is a real need to investigate alternative adaptive systems.

The FIR filter has always been the preferred implementation of the adaptive filter because it is stable and because it produces an error surface that is free of local miniums. There has always been interest in building adaptive IIR filters. [39].While not often implemented in practice there have been a number of papers published describing the implementation of the LMS algorithm to IIR filters[13]. In the early 1990's there has also been increased interest in designing adaptive continuous time filters[14]. Much of this

work is very preliminary. Suggested applications have included high performance tracking notch filters[18] and automatic tuning of integrated filters[15]. In addition there has been some unpublished work on continuous time equalizers both as automatic line build out (ALBO) systems [22] and as zero forcing equalizers [8].

This thesis explores the application of the LMS algorithm to a continuous time biquadratic filter. This is a quite generic implementation of a continuous time adaptive filter. The filter and algorithm proposed here could be applied as an interference canceller, speech processor or an equalizer. The systems identification problem was chosen as a vehicle for studying the behavior of the filter. Three different implementations are proposed. The first is a biquad with adaptive zeros only; the second is a biquad where the poles and zeros of the filter are directly adapted. The third is a biquad where the poles and zeros are indirectly adapted. The behavior of these filters is studied through simulations. The simulations are performed by mapping the filters into the discrete time domain and implementing them as computer programs. While a very general circuit implementation is shown, no specific circuit details are presented.

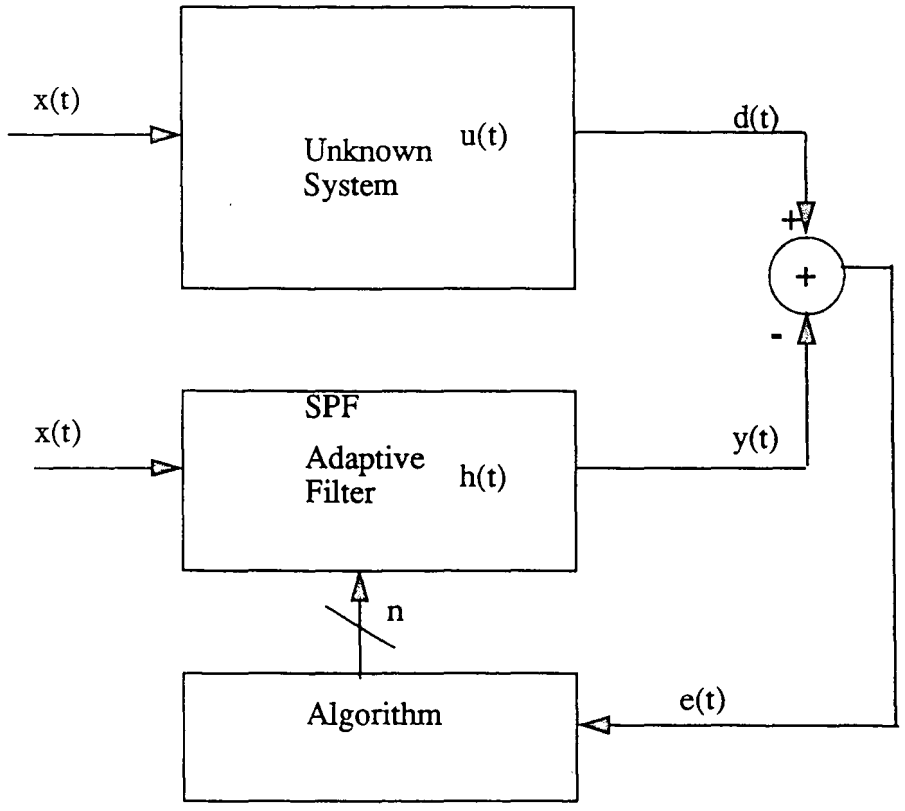
The organization of this thesis is as follows. The second chapter gives the derivation of the adaptive algorithm. The third chapter shows how the circuit could be implemented. The fourth chapter discusses how the simulator was designed and the fifth chapter presents the results. Finally some ideas for future work are presented.

2.0 Derivation of the Adaptive Algorithms

2.1 Weiner Hopf Equation

We begin the derivation of the adaptation algorithm for the continuous time adaptive filter by considering the optimal time domain solution for the system identification problem shown in Figure 1. In this problem we wish to model the unknown system u(t) with the adaptive system, h(t). The adaptive system consists of two parts, the signal path filter (SPF), and the algorithm. We require that x(t) be a zero mean, stationary signal.

FIGURE 1. System Identification Problem



To the extent that our system is successful, the optimal solution y(t) will be the best estimate of d(t) that is possible. The error for this system is the difference between the desired response and the actual response.

$$e(t) = d(t) - y(t) \quad \text{(EQ 1)}$$

$$e(t) = d(t) - \int_{-\infty}^{\infty} h(\tau)\, x(t-\tau)\, d\tau \quad \text{(EQ 2)}$$

We wish to find $h_{opt}(t)$ such that the mean squared error, $MSE = E(e^2(t))$, is minimized, where "E" represents the statistical expectation operator. One way to solve this problem is to invoke the principle of orthogonality which states that for the optimal solution, when the MSE is minimized, the input, x(t), and the error, e(t), are orthogonal [10][1][5].

$$E[e(t)\, x(t-\alpha)] = 0 \quad \text{(EQ 3)}$$

$$E\left[\left[d(t) - \int_{-\infty}^{\infty} [h_{opt}(\tau)\, x(t-\tau)\, d\tau]\right] x(t-\alpha)\right] = 0 \quad \text{(EQ 4)}$$

The expectation operator is linear and can be brought inside the integral.

$$E[d(t)\, x(t-\tau)] - \int_{-\infty}^{\infty} [h_{opt}(\tau)\, E[x(t-\tau)\, x(t-\alpha)]]\, d\tau = 0 \quad \text{(EQ 5)}$$

Recognizing the auto and cross correlation functions we write:

$$R_{dx} = \int_{-\infty}^{\infty} h_{opt}(\tau)\, R_{xx}(\alpha-\tau)\, d\tau \quad \text{(EQ 6)}$$

Equation 6 is known as the non-causal Weiner-Hopf equation for the optimum linear estimator[1]. Non-causal simply refers to the fact that it requires integration over all of time. The correlation function and the power spectral density functions are Fourier Transform pairs. Taking the Fourier Transform of both sides we have:

$$P_{dx}(\omega) = H_{opt}(\omega)\, P_{xx}(\omega) \quad \text{(EQ 7)}$$

$$H_{opt}(\omega) = \frac{P_{dx}(\omega)}{P_{xx}(\omega)} \quad \text{(EQ 8)}$$

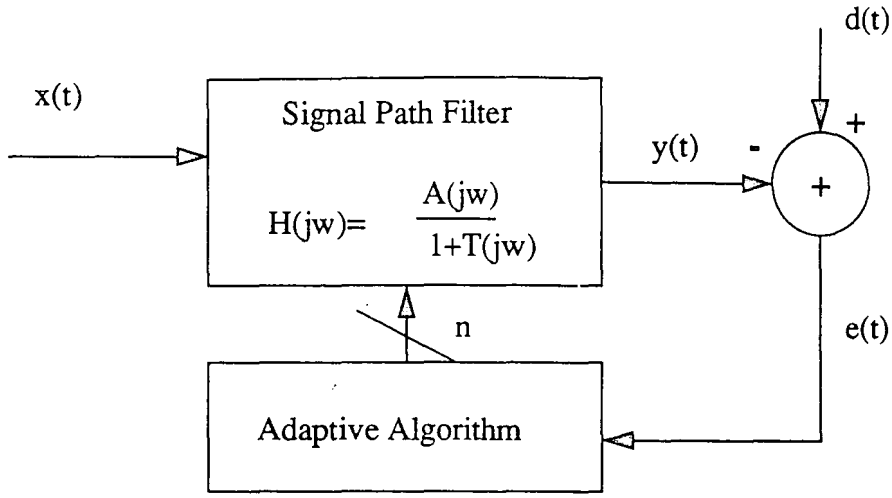
Which is the non-causal transfer function of the impulse response of the filter that is the optimal estimator of d(t). While we cannot realize this filter we can approximate it by estimating the power spectral densities of the auto and cross correlation functions of the input and desired signals of the filter.

This is not the only method of deriving the Weiner -Hopf equations. It is also possible to derive equations 6-8 directly by minimizing the mean square error [5] rather than by invoking the principle of orthogonality. For the purposes of this thesis it is simply important to recognize that the Weiner-Hopf equations are the basis for the development of these adaptive filters. Most linear discrete time filters are based upon the development of various algorithms to estimate the solution of the Weiner-Hopf equation in real time.[34] While most of the literature has utilized the simplicity of the linear transversal filter[2] a good deal of work has been done on the development of algorithms for IIR[25] and even continuous time filters[17]. In the next section we will develop the optimum solution for a continuous time adaptive filter in a manner which will allow the development of a recursive algorithm for its solution.

2.2 Continuous Time Filter Optimal Solution

Figure 2 is the block diagram of the generic adaptive filter. There are two inputs, the input signal, x(t), and the reference or desired signal, d(t). There are two outputs, the filter output, y(t) and the error signal, e(t). We assume that there are sufficient degrees of freedom in the signal path filter to do a reasonable job of matching the desired response.

FIGURE 2. General Adaptive Filter



We begin the derivation of the optimal solution for this system by assuming that all signals are zero mean, stationary signals with existing Fourier Transforms. We first take the Fourier transform of all pertinent signals so that we can work in the frequency domain. For convenience we write the transfer function of the filter in the following manner.

$$H(j\omega) = \frac{A(j\omega)}{1+T(j\omega)} \qquad \text{(EQ 9)}$$

where

$$A(j\omega) = a_0 + a_1(j\omega)^{-1} + a_2(j\omega)^{-2} + \ldots + a_{N-1}(j\omega)^{-(N-1)} \qquad \text{(EQ 10)}$$

$$T(j\omega) = t_1(j\omega)^{-1} + t_2(j\omega)^{-2} + \ldots + t_{M-1}(j\omega)^{-(M-1)} \qquad \text{(EQ 11)}$$

Writing the transfer function in this manner allows us to write:

$$Y(j\omega) = X(j\omega)\frac{A(j\omega)}{1+T(j\omega)} \qquad \text{(EQ 12)}$$

from which we can write the following recursive equation for $Y(j\omega)$

$$Y(j\omega) = X(j\omega)A(j\omega) - T(j\omega)Y(j\omega) \qquad \text{(EQ 13)}$$

The frequency domain error is:

$$e(j\omega) = D(j\omega) - Y(j\omega) \quad \text{(EQ 14)}$$

The mean squared error, J, is:

$$J = E(e(j\omega)\tilde{e}(j\omega)) \quad \text{(EQ 15)}$$

Where $\tilde{e}$ is the complex conjugate. Substituting Eq. 13 into Eq. 14 we have:

$$e(j\omega) = D(j\omega) - X(j\omega)A(j\omega) + Y(j\omega)T(j\omega) \quad \text{(EQ 16)}$$

for the error. Using this in Eq. 15 and multiplying out we have:

$$J = E[|D|^2 + |X|^2|A|^2 + |Y|^2|T|^2 - D\tilde{X}\tilde{A} - \tilde{D}XA + \tilde{D}YT + D\tilde{Y}\tilde{T} - \tilde{A}\tilde{X}Y\tilde{T} - XA\tilde{Y}\tilde{T}] \quad \text{(EQ 17)}$$

Where for convenience the $j\omega$ operator has been dropped. Taking the expectation operator inside and using the Weiner-Khintchine relation to recognize that the Fourier transform of the square magnitude of a function is the power spectral density of that function, we then write the mean square error as:

$$J = P_{DD} + P_{XX}|A|^2 + P_{YY}|T|^2 - 2A_R P_{DX} + 2T_R P_{DY} - 2|AT|P_{XY} \quad \text{(EQ 18)}$$

Where the "R" subscript refers to the real part of the quantity. Equation 18 relates the mean square error of the continuous time recursive adaptive filter to the statistical properties of the inputs and the magnitude of the transfer function.

To find the optimum solution for this equation we find the minimum of the gradient of the mean square error. To do this we take the partial derivative of Eq. 18 with respect to the numerator and denominator polynomials, A and B, and set each equal to zero.

$$\nabla J_A = 2|A|P_{XX} - 2P_{DX} - 2|T|P_{XY} = 0 \quad \text{(EQ 19)}$$

$$\nabla J_B = 2|T|P_{YY} + 2P_{DY} - 2|A|P_{XY} = 0 \quad \text{(EQ 20)}$$

We can solve for the optimal numerator and denominator coefficients if we know the statistical parameters of the input, desired response and output signals.

$$A_{opt} = \frac{T_{opt}P_{XY} + P_{DX}}{P_{XX}} \qquad \text{(EQ 21)}$$

$$T_{opt} = \frac{A_{opt}P_{XY} - P_{DY}}{P_{YY}} \qquad \text{(EQ 22)}$$

We must solve these two equations simultaneously to find the optimal solution. Once again we can only estimate this solution for real signals. Equations 21 and 22 are essentially the Weiner-Hopf equations rewritten for the filter shown in Figure 2. These forms are a little more useful in that they relate the magnitude of the numerator and denominator coefficients of the transfer function, which are quantities which we can control, to the optimal solution.

The derivation given here is very similar to the derivation given in [9] for the case of an adaptive IIR filter. For the discrete time case however, there was no need to work with Fourier transforms and the equations for the optimal solutions could be written directly in terms the correlation products. In that case the author was able to derive an adaptive algorithm directly from the optimal solutions for the coefficient vectors. There has been some work investigating adaptive continuous time filters based on power spectral estimation [4]. This work investigates real time estimates of the solutions of equations 21 and 22 by the steepest descent method.

2.3 Steepest Descent Method

The method of steepest decent is an mathematical optimization technique that is the basis for many adaptation algorithms. This family of algorithms has been used successfully for adapting both FIR and IIR filters and is the only method considered in this work. The basic algorithm itself is quite simple. A cost function for the system is designed, typically the mean-squared error, and starting at some arbitrary point the gradient for the cost function at that point is computed. Since the gradient points in the direction of the steepest growth of the function, the solution is the adjusted in the direction of the negative of the gradient. The method of steepest descent thus formulates a recursive

solution to a particular problem. It requires knowledge of the cost function for the adaptive system and the gradient of the cost function. In general we write:

$$W(n+1) = W(n) + \mu[-\nabla J_W] \qquad \text{(EQ 23)}$$

where W is the parameter being adjusted, μ, is the step size or gain and ∇J_W is the gradient of the cost function.

It is reasonable to ask how one can be sure that the algorithm will converge to the global minimum.or that it will converge at al. In general these are difficult questions to answer. For the case where the adaptive filter is an all zero or FIR filter, it is easy to show that the cost function or mean square error forms a quadratic equation[36]. The gradient must be parabolic in nature and have only one minimum. If the algorithm does converge it must converge to a global minimum. For the case of a recursive filter this is no longer true and the question of convergence to a local minimum becomes quite important[[31]. This will be discussed in a little more detail later in this chapter.

As far as the stability of the algorithm, that is will it converge at all, we know from the study of adaptive FIR filters that the steepest descent algorithm formulates a multi-variable non-linear feedback system[10]. Adaptation of a recursive filter with adjustable poles and zeros is a more complex optimization problem than the FIR filter case. Rather than trying to understand the algorithm through analytic means, we will build a model of the system and use simulations to study the behavior of the model. But before we can do that we need a real algorithm. For that we will turn to the simplest version of a gradient search technique the namely the Least Mean Squares (LMS) algorithm.

2.4 Adaptation Based on Least Mean Squares Estimate of The Error Gradient

The LMS algorithm is the standard for adaptive FIR filters. It has been also been used for adapting IIR filters with varying degrees of success[9][34][11][25]. Lately, several authors have mentioned the use of the LMS algorithm in adapting continuous time

(CT)filters.[13][14][16] It is a simplification of the steepest descent algorithm. In the method of steepest descent we needed to know the gradient of the mean square error (MSE). To know the MSE requires knowledge of the error over all time. This is obviously impractical. The LMS algorithm uses the simplest possible estimate of the MSE -- the instantaneous error. While this may seem like a poor estimate, experience has shown that in practice the LMS algorithm can give very good performance, particularly if care is taken in the selection of the step size or LMS gain μ.

We begin the derivation of the LMS algorithm for the continuous time filter by writing an expression for the error. For convenience we will work with the Laplace transform of the error

$$e(s,W) = D(s) - Y(s,W) \tag{EQ 24}$$

We estimate the mean squared error as:

$$J \approx e(s,W)\,\tilde{e}(s,W) \approx |e|^2 \tag{EQ 25}$$

We can differentiate this to get:

$$\nabla J = 2e(s,W)\frac{\partial}{\partial W}e(s) \tag{EQ 26}$$

$$\nabla J = 2e(s,W)\frac{\partial}{\partial W}(D(s) - Y(s,W)) \tag{EQ 27}$$

$$\nabla J = -2e(s)\frac{\partial}{\partial W}Y(s) \tag{EQ 28}$$

where W is whatever parameter we intend to adapt. Y(s) is a function of A(s), B(s) and the input signal, X(s). We can then write the gradients of the estimate of the MSE with respect to the parameters that are under our control.

We know that

$$Y(s) = X(s)\frac{A(s)}{B(s)} \tag{EQ 29}$$

where A(s) and B(s) are polynomials in the complex variable s.

We write the transfer function is this different but equivalent form than in Eq.9 to allow us to relate directly to standard analog filter design. The gradient equations are:

$$\nabla J_A = -2e(s)X(s)\frac{\partial Y}{\partial A} \tag{EQ 30}$$

$$\nabla J_B = -2e(s)X(s)\frac{\partial Y}{\partial B} \qquad \text{(EQ 31)}$$

In order to calculate the partial of Y(s) with respect to A(s) we need to know something about A(s). Let us assume that A(s) and B(s) are both second order functions. This implies that our active signal filter is a biquadratic section. Since the biquad is a fundamental building block for higher order filters this is a reasonable limitation[15].

If A(s) has three terms we need to calculate three partial derivatives. First we write

$$Y(s) = X(s)\left[\frac{a_0 s^2 + a_1 s + a_2}{s^2 + b_1 s + b_2}\right] \qquad \text{(EQ 32)}$$

It is straightforward to write:

$$\frac{\partial^{Y(s)}}{\partial a_0} = X(s)\left[\frac{s^2}{B(s)}\right] = \phi_{a0} \qquad \text{(EQ 33)}$$

$$\frac{\partial^{Y(s)}}{\partial a_1} = X(s)\left[\frac{s}{B(s)}\right] = \phi_{a1} \qquad \text{(EQ 34)}$$

$$\frac{\partial^{Y(s)}}{\partial a_2} = X(s)\left[\frac{1}{B(s)}\right] = \phi_{a2} \qquad \text{(EQ 35)}$$

We recognize equations 33, 34 and 35 as the highpass, bandpass and lowpass outputs of the biquad. These three equations express the sensitivity of the output signal, Y(s) with respect to each coefficient in the numerator.

We now need to calculate the sensitivity of the output with respect to the denominator coefficients. We can write:

$$Y(s) = X(s)A(s)B^{-1}(s) \qquad \text{(EQ 36)}$$

Taking the partial derivative with respect to b1 and b2 we have:

$$\frac{\partial^{Y(s)}}{\partial b1} = -X(s)\frac{A(s)}{B(s)}\frac{s}{B(s)} = \phi_{b1} \qquad \text{(EQ 37)}$$

$$\frac{\partial^{Y(s)}}{\partial b2} = -X(s)\frac{A(s)}{B(s)}\frac{1}{B(s)} = \phi_{b2} \qquad \text{(EQ 38)}$$

The first product in each is simply the output, Y(s). The second products are the same gradient filters as found in Eq. 34 and 35 We now have everything we need to put together adaptation equations for the coefficients.

It we sample and hold the continuous time outputs of the gradient filters we can write discrete time update equations for the coefficients by using the instantaneous error as an estimate of the error and the gradients of the filter output with respect to the coefficients that we calculated. This gives us:

$$a_i(n+1) = a_i(n) + \mu e(n)\,\phi_{ai}(n) \qquad i = 0, 1, 2 \qquad \text{(EQ 39)}$$

for the numerator coefficients and

$$b_j(n+1) = b_j(n) - \nu e(n)\,\phi_{bj}(n) \qquad j = 1, 2 \qquad \text{(EQ 40)}$$

for the denominator coefficients. In Eq.39, μ is the step size or gain of the LMS algorithm. In Eq. 40, ν is the step size or gain. We may or may not want to have different gains for the numerator and denominator update equations [34].

These two equations tell us how to update the filter coefficients in discrete time. While we could implement the adaptation as a discrete time process, we would prefer to have continuous time update equations. In order to translate Eq.39 and 40 into the continuous time domain we recognize that they can be thought of as forward Euler approximations of the integral. We write:

$$a_1(t) = \int_0^t \mu e(\tau)\,\phi_1(\tau)\,d\tau \qquad i = 0, 1, 2 \qquad \text{(EQ 41)}$$

$$b_j(t) = \int_0^t -\nu e(\tau)\,\phi_j(\tau)\,d\tau \qquad j = 1, 2 \qquad \text{(EQ 42)}$$

One important point is that we probably want to implement the LMS integrators as lossy integrators to help prevent and reduce problems with dc offsets [14] and coefficient saturation[27].

It should be mentioned that there is no reason that the filters proposed here cannot be implemented in the discrete time domain with switched capacitors. Similar implementations have been proposed in the past[20]. For the purposes of this work we will confine ourselves to continuous time filters.

The adaptation method proposed in this section adapts both the poles and the zeros of a continuous time filter by using the LMS algorithm. In this algorithm we use the instantaneous error as an estimate of the mean square error. We have derived expressions for the gradient of the error as functions of the filter coefficients and found that, for the numerator coefficients, these are the states of the filter. For the denominator these gradients are the states of a copy of the filter operating on the output of the signal path filter. We have shown that the numerator states are readily available in a standard biquadratic filter. For the denominator coefficients, a second copy of the filter is required. It is not necessary to update all of the coefficients. It is likely that many potential applications of continuous time adaptive filters will not require the adaptation of both poles and zeros. In the case where only the zeros of a filter need to be tuned, for example an amplitude equalizer, this algorithm would be very effective. For the case where it is desired to adapt both poles and zeros then the issue of filter stability becomes important.

2.5 Convergence and Stability Issues

The convergence and stability characteristics of the LMS algorithm when applied to linear FIR filters are well known[2][7][10]. While the system is non-linear is it possible through basic analysis to derive boundaries for values of the LMS gain, μ, for which the algorithm is stable. In addition, it can be shown that the LMS gain controls the rate of adaptation of the algorithm and the amount of excess error in the system after adaptation.

For FIR filters it is also easy to show that the equation for the MSE is quadratic implying that there are no local minima. This rather important observation implies that when the algorithm converges to a minimum then it is the global minimum.

For recursive filters all of these nice properties go away. It is no longer particularly easy to directly link the LMS gain to the stability of the algorithm analytically. In addition, the error surface is no longer quadratic [29][31]. The implications are that adaptive recursive filters are much more difficult to use than non-recursive filters. This has impeded the application of adaptive recursive filters to real world problems.

We will deal with the first analytical complexity by studying the convergence and stability properties of the filter via computer simulations. While not as satisfying as an analytical solution this is a very practical approach. Hopefully we will be able to link our empirically observations to the analytical solutions that have been developed for adaptive FIR filters. The non-quadratic error surface is more difficult to deal with. In general, since it is known that local error miniums do exist for recursive filters [25] it is difficult to guarantee that the algorithm will not become 'trapped'. A few observations may help:

1. There is evidence to support the theory that if the filter has sufficient poles and zeros, sufficient being taken as equal to or greater than the system being modeled, then the error surface will be quadratic and the filter will converge on the global minimum.[34]
2. There is evidence to suggest that in the limit, the filter will always converge to the global minimum. [23]
3. There is also evidence to suggest that the proper initialization of the filter states will reduce the impact of local miniums. [35].

In light of the complexity of the adaptation problem it is interesting to study the error for the simplest case of an adaptive CT filter. We will assume a system identification problem of the kind that we have been discussing with both the unknown system and the adaptive system first order. We will allow the adaptive filter to change its pole and zero. We can write the desired response as:

$$D(j\omega) = X(j\omega)\left[\frac{1+j\omega a1}{1+j\omega b1}\right] \quad \text{(EQ 43)}$$

The output of the adaptive filter is:

$$Y(j\omega) = X(j\omega)\left[\frac{1+j\omega a2}{1+j\omega b2}\right] \quad \text{(EQ 44)}$$

The system error is the difference between Eq. 43 and Eq. 44:

$$e(j\omega) = X(j\omega)\left[\frac{\omega^2(a2b1-a1b2)+j\omega(a1-a2+b2-b1)}{1+\omega^2 b1b2+j\omega(b1+b2)}\right] \quad \text{(EQ 45)}$$

The MSE is the average of:

$$J = |X|^2\left[\frac{\omega^4(a2b1-a1b2)^2+\omega^2(a1-a2+b2-b1)^2}{(1+\omega^2 b1b2)^2+\omega^2(b1+b2)^2}\right] \quad \text{(EQ 46)}$$

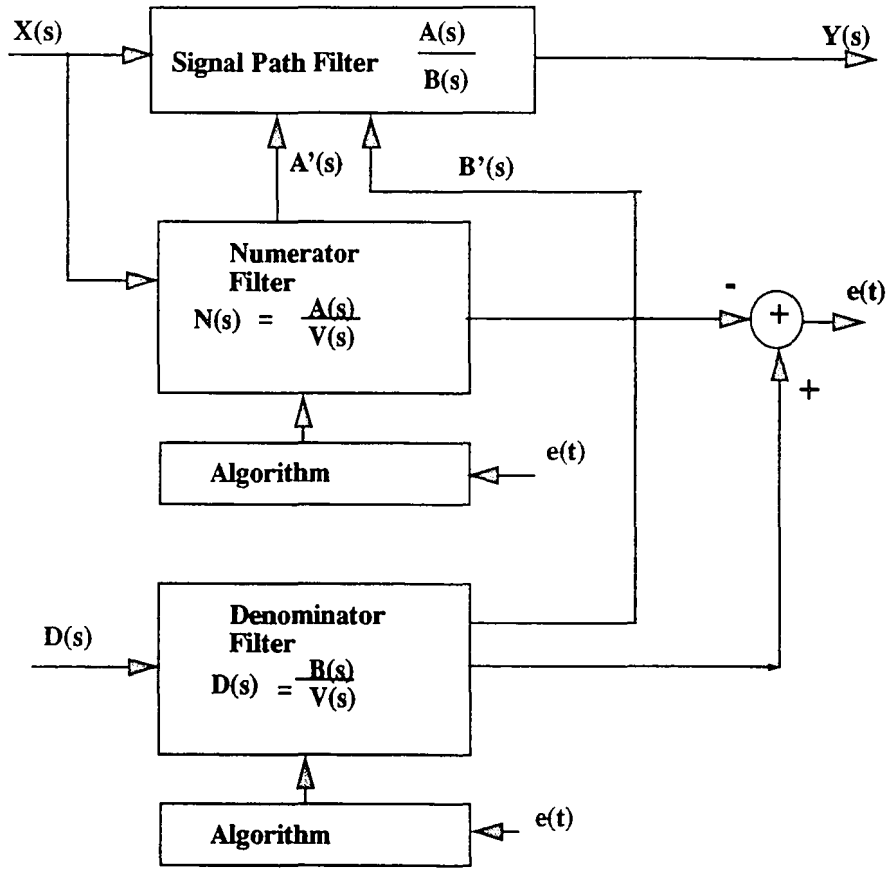
From this we see that in this simple, noiseless case, when the poles and zeros are perfectly matched,(a1=a2; b1=b2), then the error and the mean square error are zero. Furthermore, for the simple case where we adapt zeros and assume that the poles are perfectly matched, then the MSE is a quadratic. From this we know that the error surface is parabolic and we can converge to a global minimum. When we adapt both poles and zeros then the error equation is no longer quadratic. We cannot assume that we do not have local minima. It is also interesting to note that if we are adapting zeros as part of the system identification problem and we have mis-adjusted our poles the filter will still converge, but we will not, of course, be identifying the system.

From the above discussion we realize that adapting the poles of a filter is far riskier than just flirting with filter instability -- even for the simplest case. There is, however, another method of adapting IIR filters that may be quite useful for adapting continuous time filters. It is called the equation-error formulation[29].

2.6 Equation Error Adaptation

Figure3 shows the equation error formulation of an adaptive filter for the system identification problem.

FIGURE 3. Equation Error Formulation



Weights copied from N(s) and D(s) to SPF(s)

To understand how this works we first write

$$Y(s) = X(s)\frac{A(s)}{B(s)} \qquad \text{(EQ 47)}$$

$$B(s)Y(s) = X(s)A(s) \qquad \text{(EQ 48)}$$

If we let Y(s) be the best estimate of D(s) and replace Y(s) by D(s) in Eq. 48, then we can write the error for this system as:

$$e(s) = D(s)\frac{B(s)}{V(s)} - X(s)\frac{A(s)}{V(s)} \qquad \text{(EQ 49)}$$

If we minimize the mean square of Eq.49 we are implicitly minimizing the mean square system error. Since we cannot realize A(s) and B(s) by themselves,we use a dummy filter denominator, V(s), to make the numerator and denominator filters realizable. We can use the method developed previously to adapt the zeros of these filters and then copy the output coefficients to the signal path filter. There are several advantages to this method.

1. Since the poles of the signal path filter are adapted indirectly it may be simpler to check the stability of the filter.
2. For the discrete time case it can be shown that the mean square error of this system in a quadratic thus implying that there are no local miniums. It is not immediately clear that this is true for the continuous time case.

To study the convergence properties of the equation error formulation we once again turn to the simplest case of first order filters. In this case we assume that our adaptive filter has both a movable pole and a movable zero. Our numerator $N(j\omega)$ and denominator$M(j\omega)$ filters are also first order:

$$N(j\omega) = X(j\omega)\left[\frac{1+j\omega a2}{V(j\omega)}\right] \qquad \text{(EQ 50)}$$

$$M(j\omega) = D(j\omega)\left[\frac{1+j\omega b2}{V(j\omega)}\right] \qquad \text{(EQ 51)}$$

After writing out the mean square error and simplifying we find:

$$J = \frac{1}{|V|^2}\left[|X|^2(1+(\omega a2)^2) + |D|^2(1+(\omega b2)^2) - 2|XD|(1+\omega^2 a2b2)\right] \qquad \text{(EQ 52)}$$

Where V(jw) is the arbitrary denominator. This is a quadratic equation and implies that the equation error formulation should have only a global minimum. There are, however some caveats:

1. We must be careful in choosing $V(j\omega)$ so that we don't obscure information in the error.

2. There must be access to a reliable desired signal. While we have so far always assumed that d(t) was available, in many potential applications such as equalizers there is no desired signal. In these cases the equalizer output is often used as d(t) leading to what is known as decision directed operation. This may make the equation error formulation unworkable for these applications.

3. It is known that the discrete time adaptive filters built on the equation error formulation can suffer from coefficient bias. This condition is apparently the result of the filter attempting to jointly identify the system poles and simultaneously minimize the error noise power. Coefficient bias is known to be a function of the input signal to noise ratio. Continuous time equation error filters will suffer from the same problem. [29]

2.7 Variations of the Algorithm

There are a number of variations of the LMS algorithm that have been proposed and applied with varying degrees of success[2]. Most of the variations apply some sort of performance criteria to the error estimate. This can take the form of filtering the instantaneous error in an attempt to produce a better estimate of the MSE or using the absolute value or the sign of the error to simplify the computations. Often the algorithm is normalized to reduce the convergence sensitivity to the input power level. For the purposes of this thesis, only the simplest case of the LMS algorithm was considered.

2.8 Summary

In this section we have shown the optimum Weiner solution for a continuous time filter used in a system identification problem. We have derived the optimal solution for an analog continuous time filter as a function of the input and output power and cross power spectral densities. We have discussed the application of the steepest descent algorithm and have derived adaptive equations based on the LMS algorithm for a continuous time biquadratic filter. We have shown how this filter can be adapted in the following manners:

1. Adaptive Zeros Only
2. Adaptive Zeros and Poles in the Output Equation Formulation

3. Adaptive Zeros and Poles in the Equation Error Formulation.

We have briefly studied the stability and error performance of these algorithms by using simple first order filters and have discussed some of the pros and cons of each method of adaptation. In the next section we will look at some potential circuit implementations of these filters at the simple block diagram level.

3.0 Circuit Models for Adaptive Biquad

3.1 State Variable Filter With Adaptive Zeros

Now that we have developed a mathematical model showing how we can update the coefficients for a continuous time biquad, let us look at a potential circuit implementation for such a filter. Referring back to Figure 2, we see that we need three components:

1. Signal Path Filter
2. Error Processing Block
3. Adaptation Circuit (both numerator and denominator coefficients are to be adapted.

Figure 4 shows the block diagram for a general purpose adaptable state variable biquadratic filter made with an inverting integrator.This filter realizes the function:

$$H(s) = \frac{a_0 s^2 - a_1 \omega_o s + a_2 \omega^2_0}{s^2 + b_1 \omega_o s - b_2 \omega_o^2} \qquad \text{(EQ 53)}$$

To make this filter into a standard fixed biquad we require that $b_2 = -1, b_1 = \frac{1}{Q_p}$.

In this filter both the poles and zeros can be adapted by changing the multiplier coefficients. For the case of the numerator coefficients this is straight forward. The outputs of the LMS integrators are the multiplier coefficients for each term. Updating is more difficult for the denominator terms because of the non-zero mean values. For example, we may initialize the filter with b2 = 1 and b1 = -1/Q. When we let the filter adapt we perturb b1 and b2 around these values e.g.

$$b2(t) = -1 + \delta b2(t) \qquad \text{(EQ 54)}$$

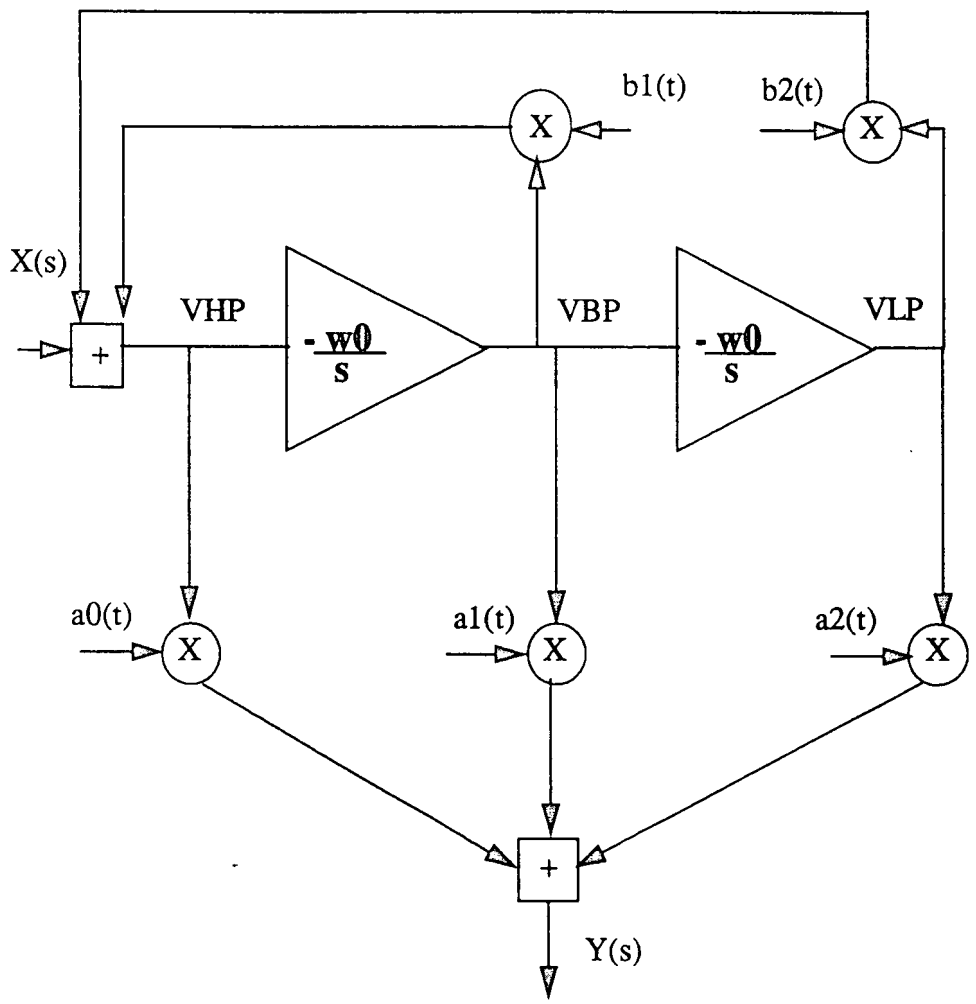
and

$$b1(t) = \frac{1}{Q} + \delta b1(t) \qquad \text{(EQ 55)}$$

where the preturbation terms are the difference between the output of the LMS integrators and the fixed terms. We must pay attention to the normalization of these terms with respect to the pole frequencies and with respect to the signs. It is reasonable to expect that this can be done as part of the processing that checks the stability of the poles. It is recognized that

this is not necessarily an easy circuit problem and the details of the solution for this problem are beyond the scope of this work. To simplify the analysis we will work with a normalized frequency of 1 rad/sec. The circuit shown in Figure 4 is generic and

FIGURE 4. State Variable Adaptive Biquad

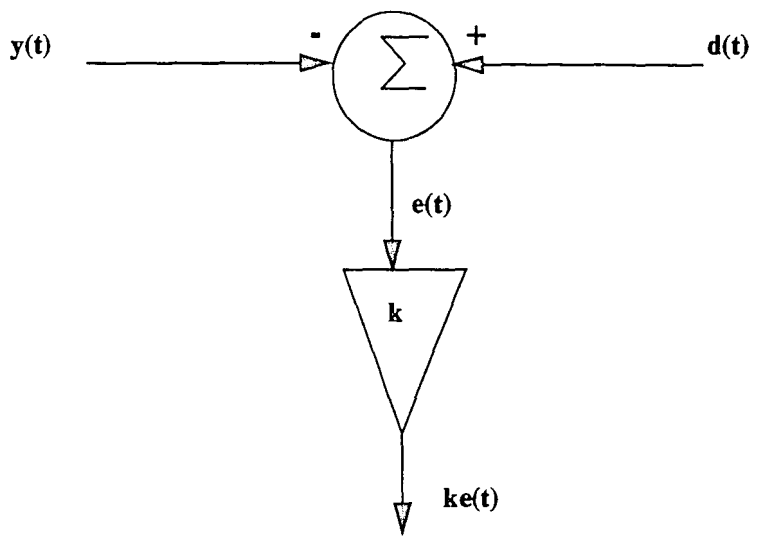


works for all three adaptive filter cases. When we are going to adapt only the zeros then we can fix the recursive multiplier inputs or replace the multipliers with fixed gain elements. For the case when we are adapting based on the output error method, then the LMS integrator outputs drive the numerator multipliers through some normalizing gain. The denominator multipliers must first be checked for stability and normalized before being passed to the multipliers. The same circuitry is used for the equation error method of

adaptation with the multiplier inputs coming from the equation error filters instead of the LMS integrators.

The second part of the filter, the error processing circuit, is shown in Figure 5.

FIGURE 5. Error Signal Processing

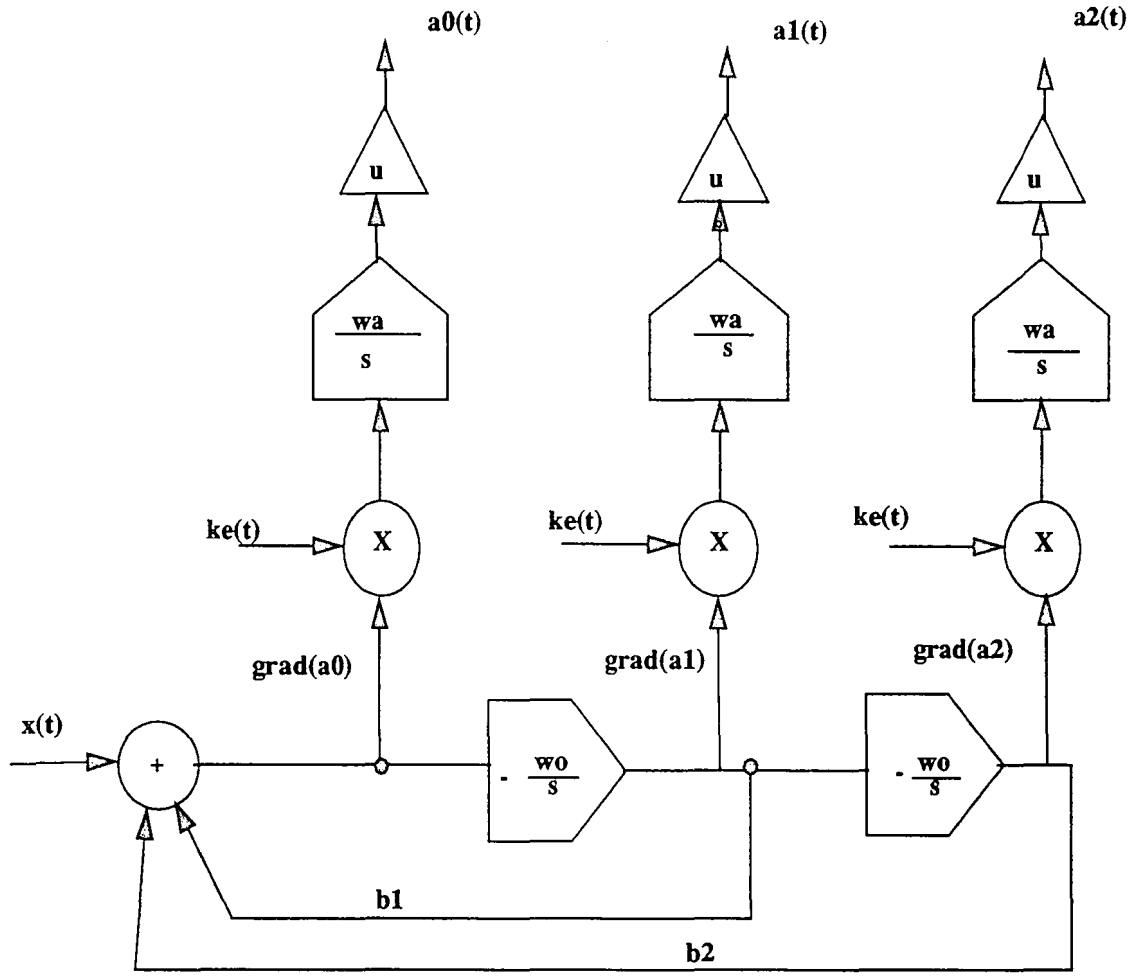


The factor of k is not the error loop gain, μ, but is in addition to that gain. Its purpose is to reduce the sensitivity of the error loop to dc offsets in the LMS integrators [14]. For simulation purposes it is taken to be 1. However, for real circuit applications this gain can be quite critical. Any additional error signal processing, e.g. filtering, taking the sign, etc., is done in this block.

For the equation error formulation the inputs to the error processing circuit become the output for the numerator filter for y(t) and the output of the denominator filter for d(t). In all other respects the circuit is the same.

The simplest case under consideration is the case where we adapt only the zeros of a CT biquadratic filter. In that case we only need one more circuit function. We need to calculate the gradients for each coefficient and generate the coefficients with LMS integrators. Figure 6 shows the circuit block to realize these functions.

FIGURE 6. LMS Adaptation Circuit and Gradient Filter



In Figure 6 we see that the recursion loop formed by the two integrators and gain paths b1 and b2 is identical to the states of the signal path filter. We recognize grad(a0), grad(a1) and grad(a2) as VHP, VBP and VLP in Figure 4. It is not necessary to build a separate filter to compute the gradients. Instead we use the states of the signal path filter itself. The requirement that all three filter outputs be available simultaneously limits the filter structure somewhat[30].

The gradient outputs are multiplied by the amplified error signal, ke(t), and integrated. While a more formal proof is available in [14], it is easy to see how amplifying the error signal before the integrator will reduce the overall sensitivity to dc offsets. It is

critical that LMS integrators have very low offset since any error here will contribute to the final mis-adjustment of the filter. In addition, the use of leaky integrators may help reduce problems with coefficient saturation and wandering [27]. In the full gradient adaptation algorithm the time constants of the LMS integrators must be less than the highest frequency component of the error signal or else the frequency response of the integrator will effect the adaptation process and possibly impact the loop stability. This matter will be discussed in more detail later in this work.

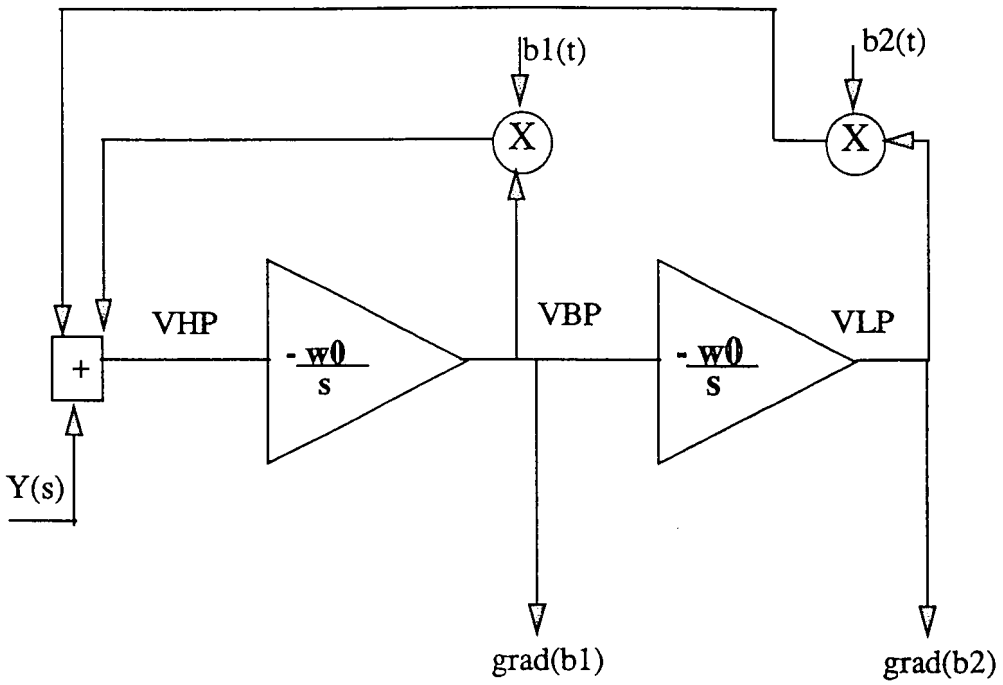
Finally the outputs of the LMS integrators are scaled by the LMS gain, μ. In the case where $k \neq 1$, then μ has to be scaled by $\frac{1}{k}$ to maintain the same loop dynamics. In addition, as will be seen later, in a real filter it is probably necessary to normalize the value of μ by the gradient filter gain. This will help maintain loop stability independent of the input signal power.

Using the filter to provide its own gradients provides self normalization of the gradients

3.2 State Variable Filter With Adaptive Poles and Zeros

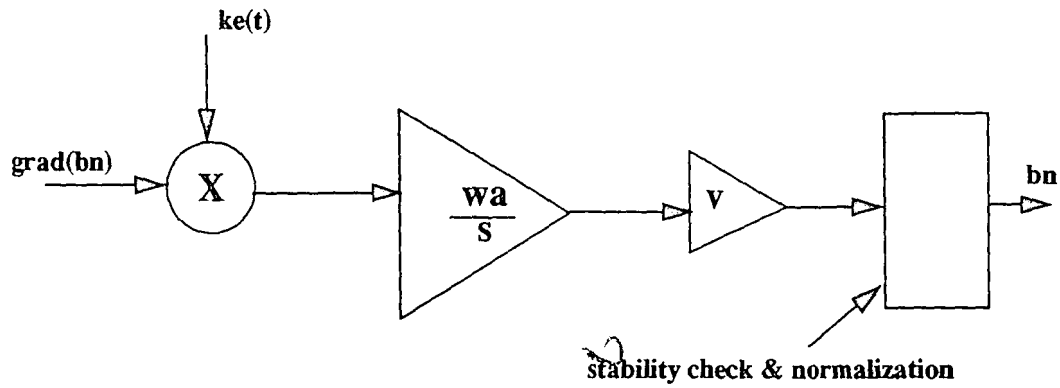
In the case where we want to adapt both the poles and the zeros of the signal path filter shown in Figure 4, we need to generate the gradients for the denominator coefficients. We know how to do this from equations 37 and 38. We need to build another filter which is a copy of the filter path signal and use it to filter the output of our signal path filter. Figure 7 shows the structure to do this.

FIGURE 7. Numerator Gradient Filter



The denominator coefficients for this filter must track the denominator coefficients for the signal path filter. There is, or course, no need for the output summing elements; in all other respects this filter is identical to the signal path filter. The LMS processing for one coefficient is shown in Figure 8. The last element is the yet undesigned circuit that checks for the stability of the poles and normalizes the value of the particular coefficient.

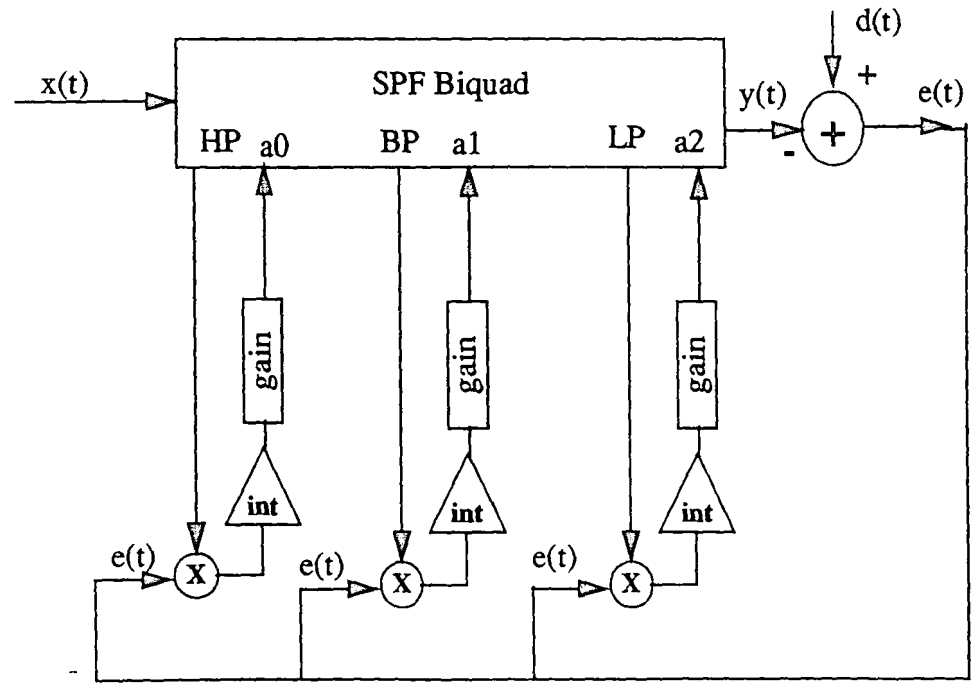
FIGURE 8. Denominator Term LMS processing



3.3 Zero Adapting Biquad

We now have all of the things we need to build any of the three types of adaptive filters proposed in this thesis. The first type, the adaptive zero only biquad, we need only a single biquad signal path filter (SPF), an error processing block, and three LMS signal processors. The full circuit is shown in Figure 9.

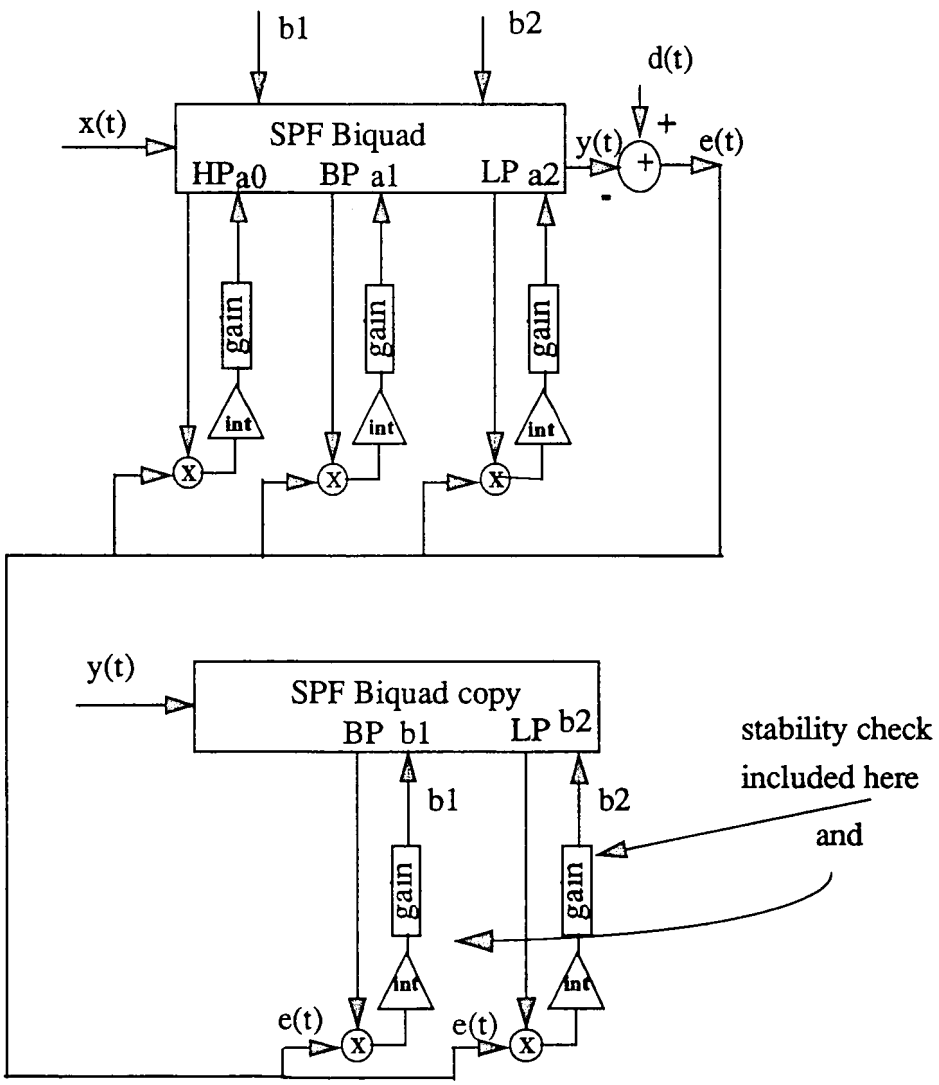
FIGURE 9. Adaptive Zero Only Biquad



3.4 Pole-Zero Adaptive Filter

To adapt both poles and zeros we need a second copy of the SPF working on the output and two more LMS signal processing blocks. Figure 10 shows the full circuit.

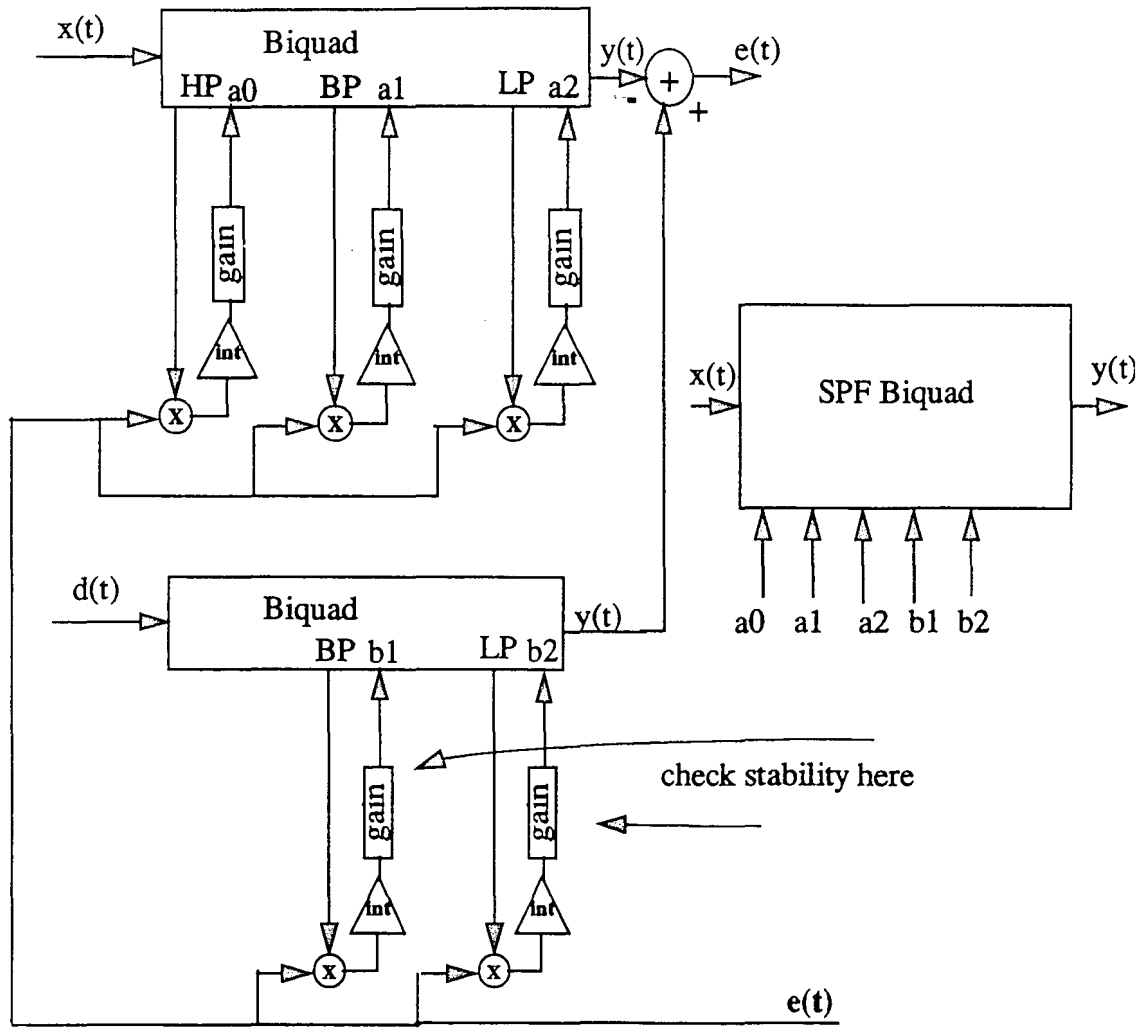
FIGURE 10. Pole-Zero Adaptive Filter



3.5 Equation Error Adaptive Filter

The third configuration is the equation error formulation. For this we need two copies of the zeros only filter and a third signal path filter. Figure 11 shows this circuit.

FIGURE 11. Equation Error Filter



3.6 System Transfer Function

In the limit, as the filter converges, the coefficients must reach a constant steady state value. This means that the average value of the product of the gradient and the error must be constant and that in the limit the signal path filter will be a well behaved second order function. However, we cannot assume anything about the transient response of the adaptation loop. This is apparent when we attempt to look at the transfer function of the filter. We have for a simple first order filter with only a single adaptive zero:

$$Y(s) = X(s)\left[\frac{s+\left[\frac{1}{s}\mu e(s)\frac{X(s)}{(s+p1)}\right]}{s+p1}\right] \qquad \text{(EQ 56)}$$

Where e(s) depends on Y(s) and X(s). Eq. 56 is too complex to easily be much help in understanding the transient behavior of the adaptation loop.

It is a little more enlightening to study the behavior of the filter coefficients. We begin by considering the first order adaptive pole and zero filter. We assume that we will use a lossy LMS integrator with a single pole at ωa. The coefficient equation for the zero is:

$$z(s) = \frac{\mu}{s+\omega a}\left[\frac{e(s)X(s)}{s+p1}\right]. \qquad \text{(EQ 57)}$$

For the pole the coefficient is:

$$p1(s) = \frac{-\nu}{s+\omega a}\left[\frac{e(s)Y(s)}{s+p1}\right] \qquad \text{(EQ 58)}$$

In Eq. 57 we assume that we can pretend that the pole is changing very slowly with respect to the zero. This corresponds to $\nu \ll \mu$ which, as we will see later is necessary in order to maintain the stability of the control loop. We see that the location zero of our first order filter will have at least a second order response Of course, this is only a rough approximation since Eq. 57 is recursive. Equation 58 is recursive in both Y(s) and p1(s); it is not clear what conclusions we can draw from this. We can however note that in both equations we must be careful where we put the integrator pole with respect to the expected range of the filter pole. For a second order system we can assume that the denominator will become third order. We must be even more aware of the integrator pole placement in that case. It seems prudent to make sure that the integrator pole frequency is much lower than the highest signal pole.

3.7 DC Sensitivity

We can take a quick look at the dc sensitivity of the LMS loop by considering the LMS update equation in the time domain. We can write

$$a(t) = \mu \int_0^t [k e(t)\,\alpha(t) + m]\,dt \qquad \text{(EQ 59)}$$

where α it the gradient signal and m is any dc offset. In the limit we know that the coefficient is a constant. If we take the expectation of the integrand in eq.59 we have

$$kE[e(t)\,\alpha(t)] = -m$$
$$E[e(t)\,\alpha(t)] = \frac{-m}{k} \qquad \text{(EQ 60)}$$

The implications here are:

1. The effect of any dc offset in the integrators is reduced by k.

2. Since the integrator has infinite gain at dc, any dc component in the signal cam cause the integrator output to ramp. Thus the requirement that all signals be zero mean.

The problems of DC offsets in analog adaptive filters are well known[26]. While there are some circuits tricks like the one given here that can help, good design practices and careful attention to the dc performance of all circuits is the best approach.

3.8 Stability Issues

When we adapt the poles of the signal path filter we have to worry about the stability of the resultant filter as well as the stability of the algorithm. This is not necessarily an easy problem to solve in real time. There is a well known and simple test for the stability of discrete time second order IIR filters based on the so called stability triangle.

FIGURE 12. Stability Triangle

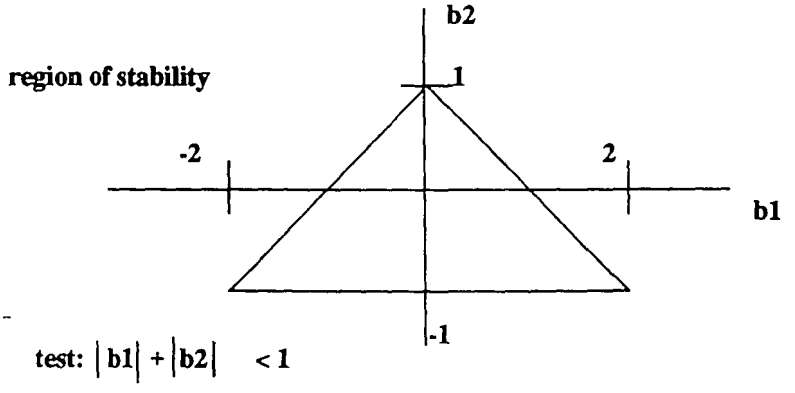


Figure 12 shows the region of stability for a biquadratic IIR filter. The stability test shown scribes a unit circuit that is a subsection of this region. It is fairly simple to apply this test

to a computer program or an IIR filter and simply limit the region within which the poles are allowed to move.

For the continuous time case this is a more difficult question. If we limit ourselves to a second order filter, then a simple test for stability of the function is to look at the sign of the b1 coefficient. If we assume complex poles then:

$$s^2 + b_1 s + b_2 = (s + P_{1R} + jP_{1i})\ (s + P_{1R} - jP_{1i}) \qquad \text{(EQ 61)}$$

We know that if the poles are to be in the left half of the s-plane then the b1 or $\frac{1}{Q}$ coefficient must be positive. As b1 approaches zero the poles are approaching the imaginary axis. We could simply limit the minimum value of b1 in an attempt to preserve stability. We also know that in the real world b2 must be a positive number. However, the LMS algorithm is not aware of this and we must take steps to prevent b2 from becoming negative.

This simple test does not work in the case where we have two real poles. In that case we can easily have one pole slightly in the RHP and the other deep in the LHP. The filter is, of course, unstable but the Q coefficient is masked by the unsymmetrical nature of the system. It is not clear if this is a real possibility. Intuitively we tend to look at the LMS algorithm as an intelligent feedback network that will move the poles in the same sort of fashion as we would observe in a root-locus construction. Thus we expect poles to start on the real axis, move together, break away from the real axis, and move towards the imaginary axis in a symmetrical path. In this case the b1 coefficient test should be an adequate test. The filter stability issue has been, and will continue to be, a major impediment to the development of CT adaptive filters. This is one reason that the CT filter with adaptive zeros only is attractive.

3.9 Summary

In this chapter we have examined some potential circuits for implementing the three types of continuous time adaptive filters derived in chapter 2. We have seen that all of the filters can be readily implemented with a standard biquadratic section with analog multipliers in place of fixed gain elements. We have discussed some of the issues that will have to be dealt with when actually designing such a filter. We have taken a cursory look at the transient behavior of the filter coefficients and learned that in adaptive filters like these it is very difficult to derive clean meaningful mathematical expressions to tell us exactly how to design. We have gotten a first order feeling for how to design the LMS integrator and for the kind of dc offset performance that we will require. Finally, we have proposed two simple stability checks one for the discrete time IIR case and one for the continuous time case that will help us check for filter stability. In the next section we will discuss the simulator that was designed to further study these problems.

4.0 Simulation Methodology

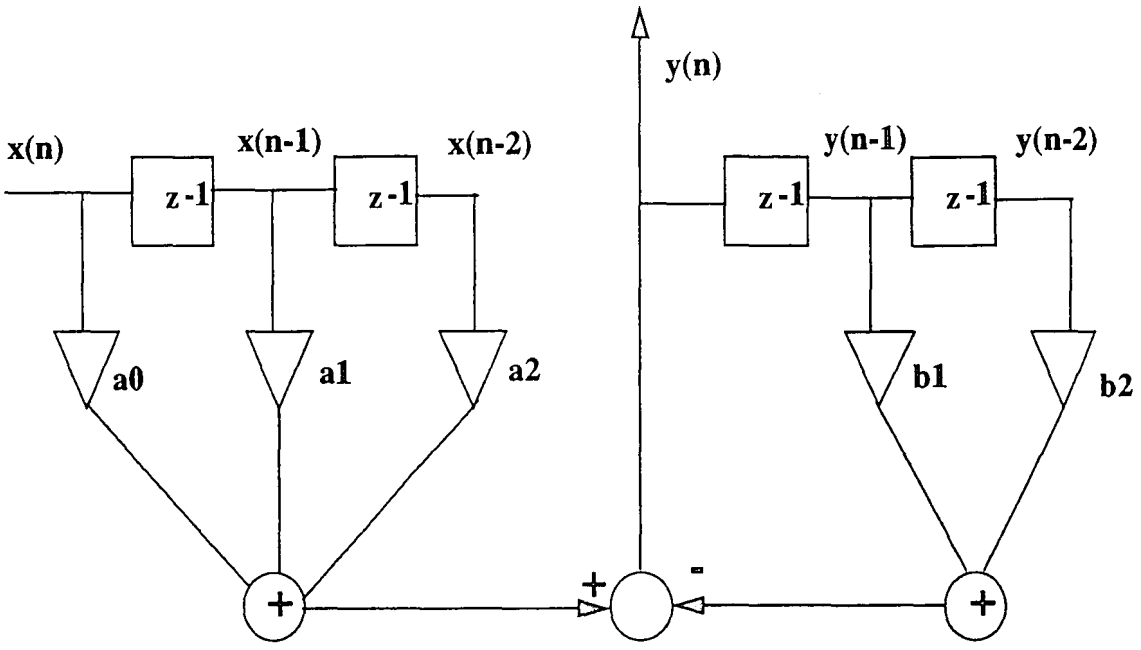
4.1 Discrete Time Filter Models

There are a number of ways in which one could simulate a continuous time adaptive filter.

1. Write a computer program to compute the complete "continuous time" solution for the filter over a range of time.
2. Map the continuous time filter into the discrete domain then design a digital filter that simulates the behavior of the analog filter.
3. Build high level models of the various components suitable for use in an electrical simulator package such as SPICE.
4. Use a canned mathematical analysis system such as MATLAB or MATHCAD.

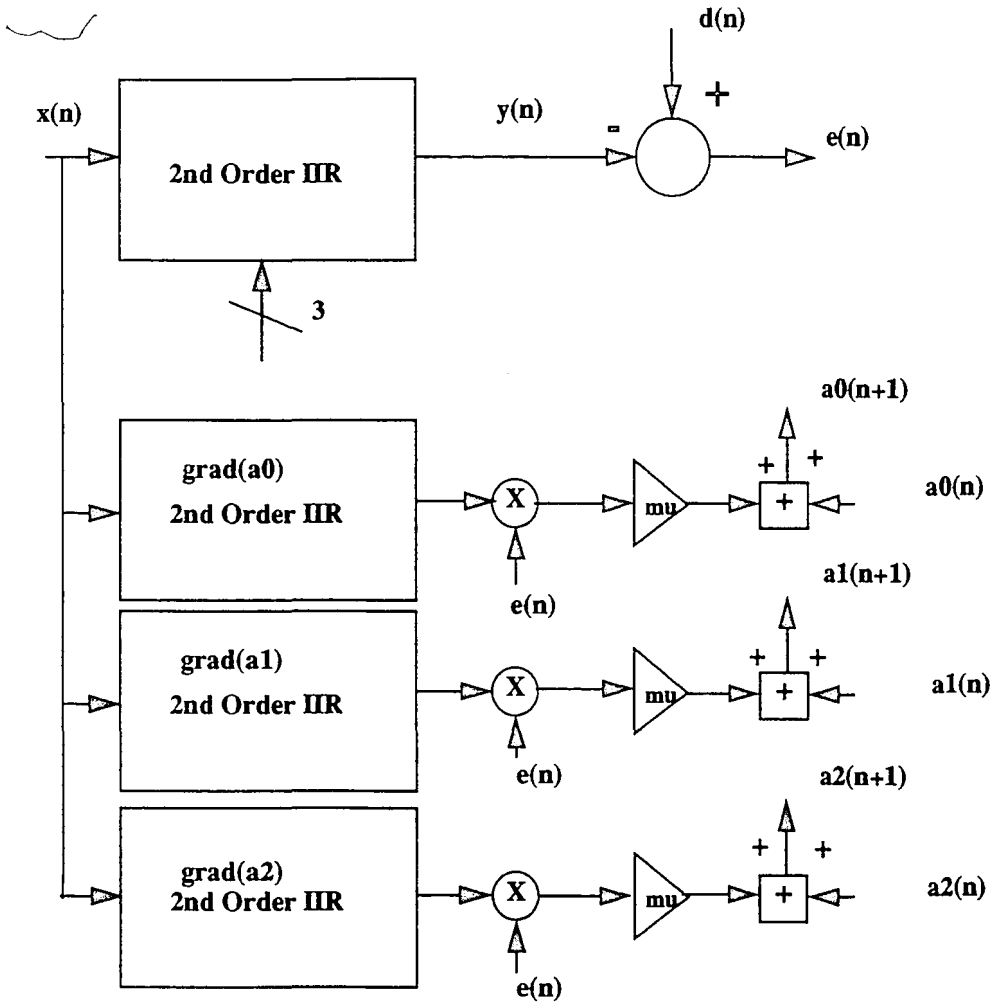
The second method was chosen for this work. Simulating continuous time systems by mapping them into the discrete domain is an widely accepted simulation technique[12]. Some care must be taken to avoid aliasing, but this is not a major impediment. What we will do is use the bilinear transform to map the continuous time filter into a discrete time filter. We can then use a computer program to run the discrete time filter on data of our creation. We will begin by looking at a block diagram for a second order IIR filter. This will be the basis for our model.

FIGURE 13. Second Order IIR Filter



Of course we also need to map the gradient filter(s) and the LMS adaptation circuits into the discrete domain. The gradient filters look exactly like the signal path filter except with different numerator coefficients.The entire simulator structure for the case when we are adapting zeros is shown in Figure 14.

FIGURE 14. Second Order Biquadratic Adaptive Filter Model (Zeros Only)



All four filters have identical poles. The zeros for the gradient filters are calculated by using the bilinear transform to map the continuous time gradient filters into the discrete domain.

In the case where we are going to move poles and zeros using the output error formulation, the simulator is more complicated. Figure 15 shows the block diagram

FIGURE 15. Second Order Biquadratic Adaptive Filter Model (Poles and Zeros)

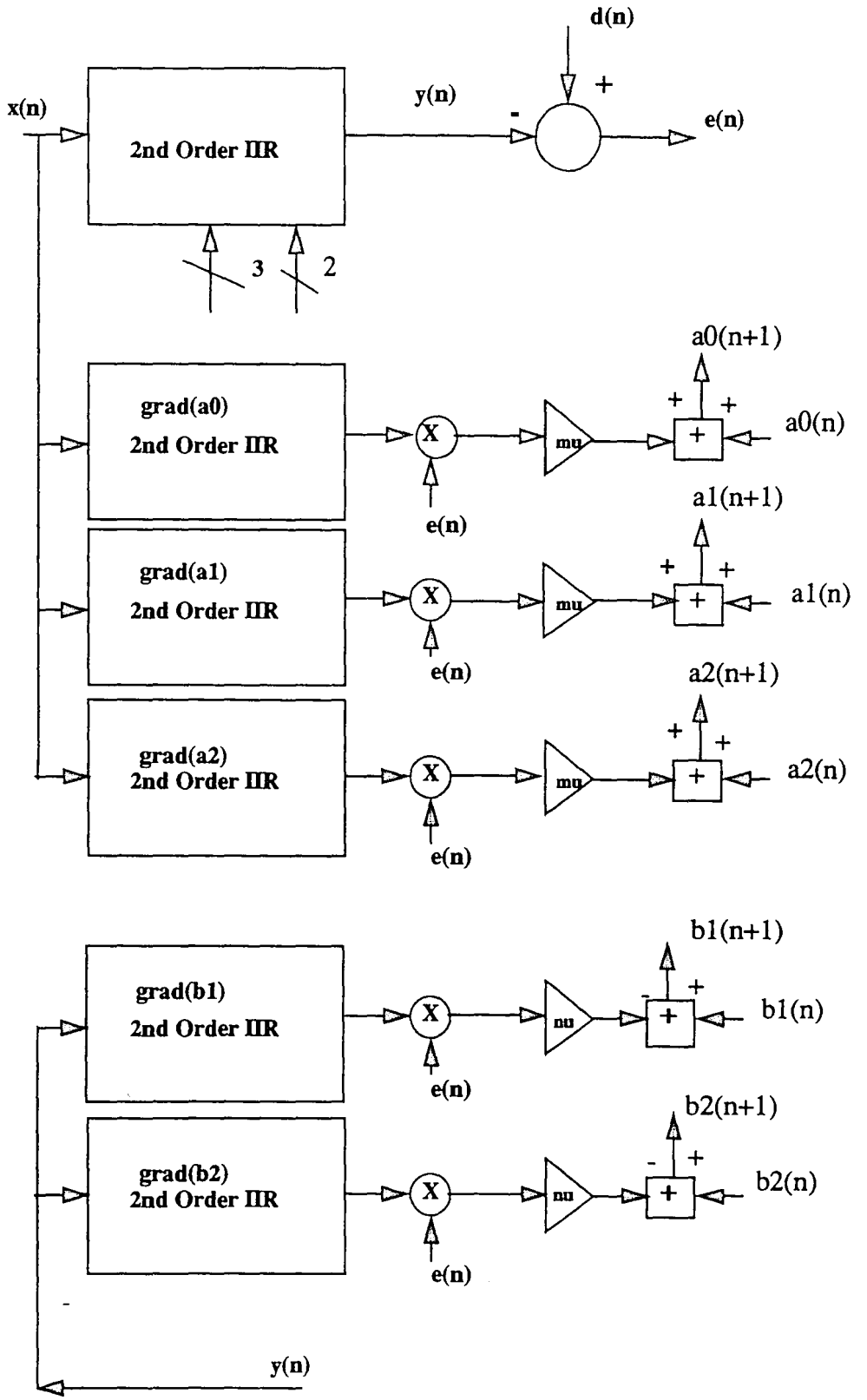


Figure 15 does not show the circuitry needed to limit the excursion of the poles or check for stability. In addition, neither figure shows any circuits to limit excursion of the zeros or to provide a leakage path for the LMS integrators. These functions are simple to include in the simulator program.

It is clear by now that the equation error formulation is simply a reconfirmation and expansion of the adaptive zeros only case. The simulator for this filter is simply an expanded version of Figure 14.

4.2 Mapping the Filter

Mapping the filter was done using the well known bilinear transform that relates points in the s-plane to points in the z-plane by

$$s = \frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) \qquad \text{(EQ 62)}$$

This transform maintains stability but warps frequencies near the band edge. When used to design digital filters from analog prototypes it is important to pre-warp critical analog frequencies the preserve the correct filter shape[28][37]. For the purposes of this work we are not interested in exact filter shapes but only in changing the filter characteristics in response to the adaptation algorithm. Thus it is not necessary to pre-warp the continuous time frequencies.

When simulating analog systems with a discrete time simulator the simulation sampling rate plays a major role in the accuracy and resolution of the simulator. On the one hand the higher the effective sample rate the more accurate the simulation becomes. On the other the higher the sample rate the more simulation time steps are necessary to encompass the same amount of time. Because of the very long time constants typically involved in LMS adaptive filters and because of the computer time requirements a standard packet length of 8192 samples was chosen to facilitate the use of the fast fourier transform for these experiments. Longer runs were accomplished by re-initializing the

filter with the final points from a previous run and passing another 8192 samples through the filter. All of the filters were designed near or with a normalized frequency of 1 rad/s. A basic normalized sampling rate of 10 Hz. was chosen giving a sampling ratio of 20π. For a sample length of 8192, this gives a frequency resolution of $\frac{10}{8192} = 1.22mHz$ which was found to be adequate. A higher sampling rate would have required a proportionally larger sample size to maintain the same resolution. It was found that the penalty in execution time was not warranted.

4.3 Input Signal Generation

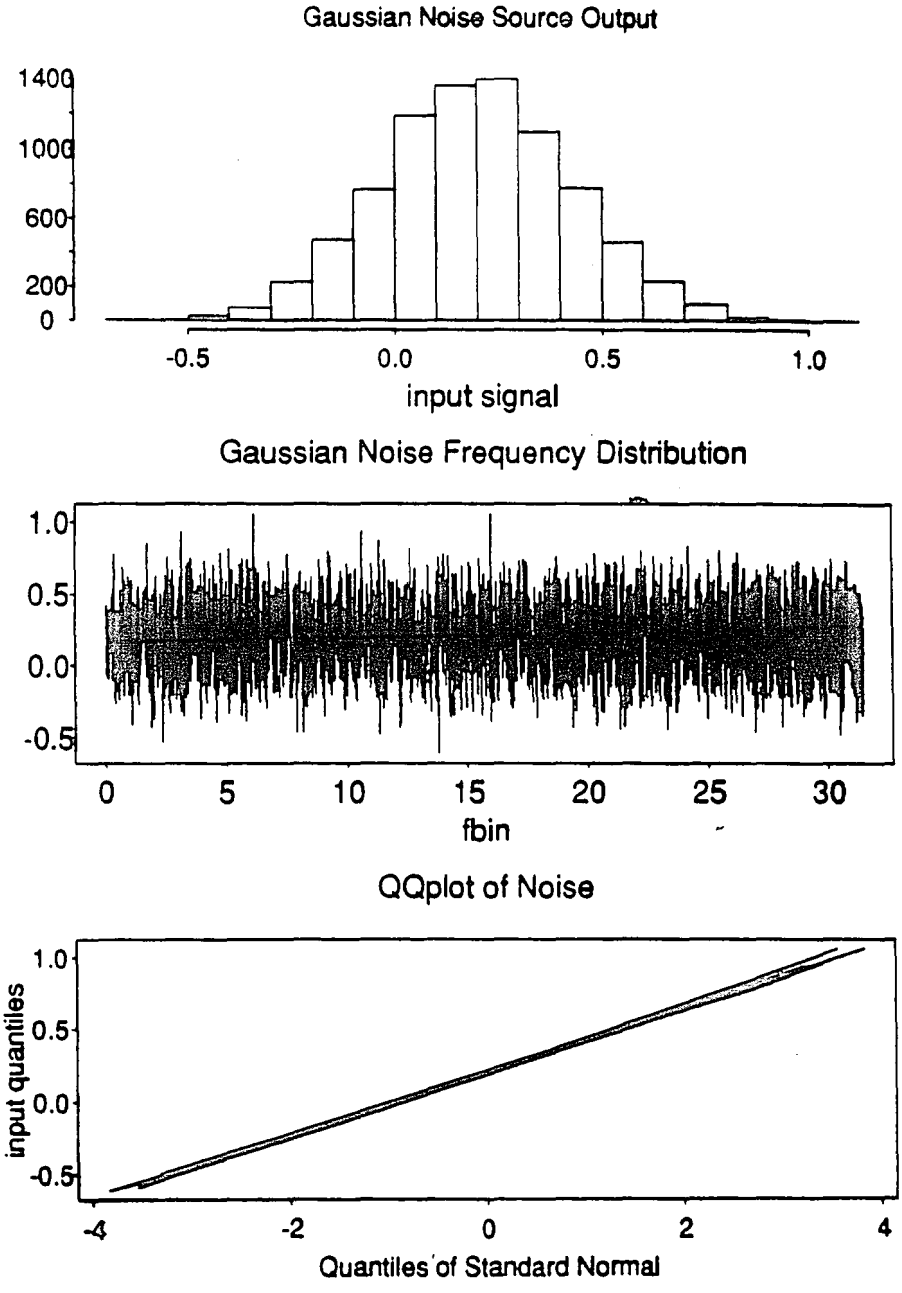
Many of the simulations performed required a zero mean gaussian noise input signal. A computer program was written to generate these by the method proposed in reference [28]. This method uses the central limit theorem that states that in the limit the sum of N identically distributed, independent, uniform random variables tends to be Gaussian. e.g.

$$y(n) = \frac{1}{N}\sum_{i=0}^{N-1} x[nN-i] \qquad \text{(EQ 63)}$$

N was chosen to be 10 for these experiments. There are a number of ways to check the resulting distribution. One can "eyeball" the flatness of the noise in the frequency domain, or the shape of the histogram or one can use a statistical technique called a quantile-quantile plot. This plots the quantiles of one statistical distribution against the quantiles of another statistical distribution [6]. For example, the mean of A is plotted against the mean of B; the upper quantile of A against the upper quantile of B. If the two distributions are identical then the plot follows the line A=B. If one of our distributions is gaussian then we can compare the distribution of our noise generator to a true gaussian distribution. Figure 16 shows all three methods for looking at the output of our noise generator. We can conclude from these that the t input noise signal is zero-mean and gaussian.

In the next section of this thesis we present some experimental results.

FIGURE 16. Input Noise Source Characteristics



5.0 Experimental Results

5.1 Continuous Time Filter With Adaptive Zeros

The first set of experiments were conducted on the filter with adaptive zeros and fixed poles.This is the simplest case and the most easily implemented in practice. For these experiments a prototype second order filter was designed and used as part of a system identification problem. Various experiments were conducted to investigate the behavior of the filter with variations in the input signal power and LMS gain, with both correlated and uncorrelated input signals, with dc offsets in the signal and with variations in the input signal to noise ratio.

5.1.1 Biquadratic Gain Equalizer

For these experiments a biquadratic gain equalizer was chosen as the unknown system. This choice was made for the following reasons

1. It requires adaptation of both zeros
2. It represents a potential application of a real filter

The biquadratic amplitude equalizer function is well known:

$$H(s) = K\left(\frac{s^2 + \frac{\omega_z}{Q_z}s + \omega_z^2}{s^2 + \frac{\omega_p}{Q_p}s + \omega_p^2}\right) \qquad \text{(EQ 64)}$$

where K = a0 and controls the overall gain. The low frequency gain is $\frac{\omega^2}{\omega_p^2}K$. The high frequency gain is K. If $\omega_z = \omega_p$ then the function is symmetrical and the maximum or minimum gain is at the pole frequency and is $\frac{Q_p}{Q_z}K$. This function is often used as a variable amplitude equalizer with the adjustment being made by changing Q_z. This filter can also be used as a delay equalizer by constraining the numerator and denominator coefficients to be equal except for $\frac{\omega_z}{Q} = -\frac{\omega_p}{Q_p}$.

The amplitude equalizer transfer function to be matched was chosen as:

$$0.25\left(\frac{s^2+5.6126s+1}{s^2+1.403s+1}\right) \qquad \text{(EQ 65)}$$

which has a pair of complex poles and two real zeros. Figure 17 shows the frequency response of this filter after it has been mapped into the discrete domain. The sampling frequency for the bilinear transform was 10. Table 1 gives the filter parameters.

TABLE 1. Gain Equalizer Parameters

A(s)	a0 = 0.25	a1 = 1.403	a2 =0.25
A(z)	a0 = 0.299049	a2 = -0.46497	a2 = 0.16825
B(s)	b0 = 1.0	b1 = 1.403	b2 = 1
B(z)	b0 = 1.0	b1 = -1.85988	b2 = 0.869252
zeros(s)	z1 = -0.184	z2 = -5.4277	
zeros(z)	z1 = 0.9817	z2 = 0.5730	
poles(s)	p1 = -0.70155	+j0.71261	
	p 2 = -0.70155	-j0.71261	
poles(z)	p1 = 0.9299	+ j0.06643	
	p2 = 0.9299	-j0.06643	
A(initial)	a0 = 1	a1 = -.6	a2 =.05
zeros(init)	z1 = 0.1	z2 = 0.5	

We began by generating the output response of this filter to a zero mean white gaussian noise input vector 8192 samples in length. We then used this output signal as the desired signal input to the adaptive filter. The WGN signal was used as the input. For this first experiment we had perfectly adjusted poles. The initial zeros were arbitrarily chosen as shown in Table 1. The parameters for the three gradient filters are given in Table 2.

Figure 18 shows the input and desired signals in the frequency domain.

FIGURE 17. Prototype Filter Transfer Function

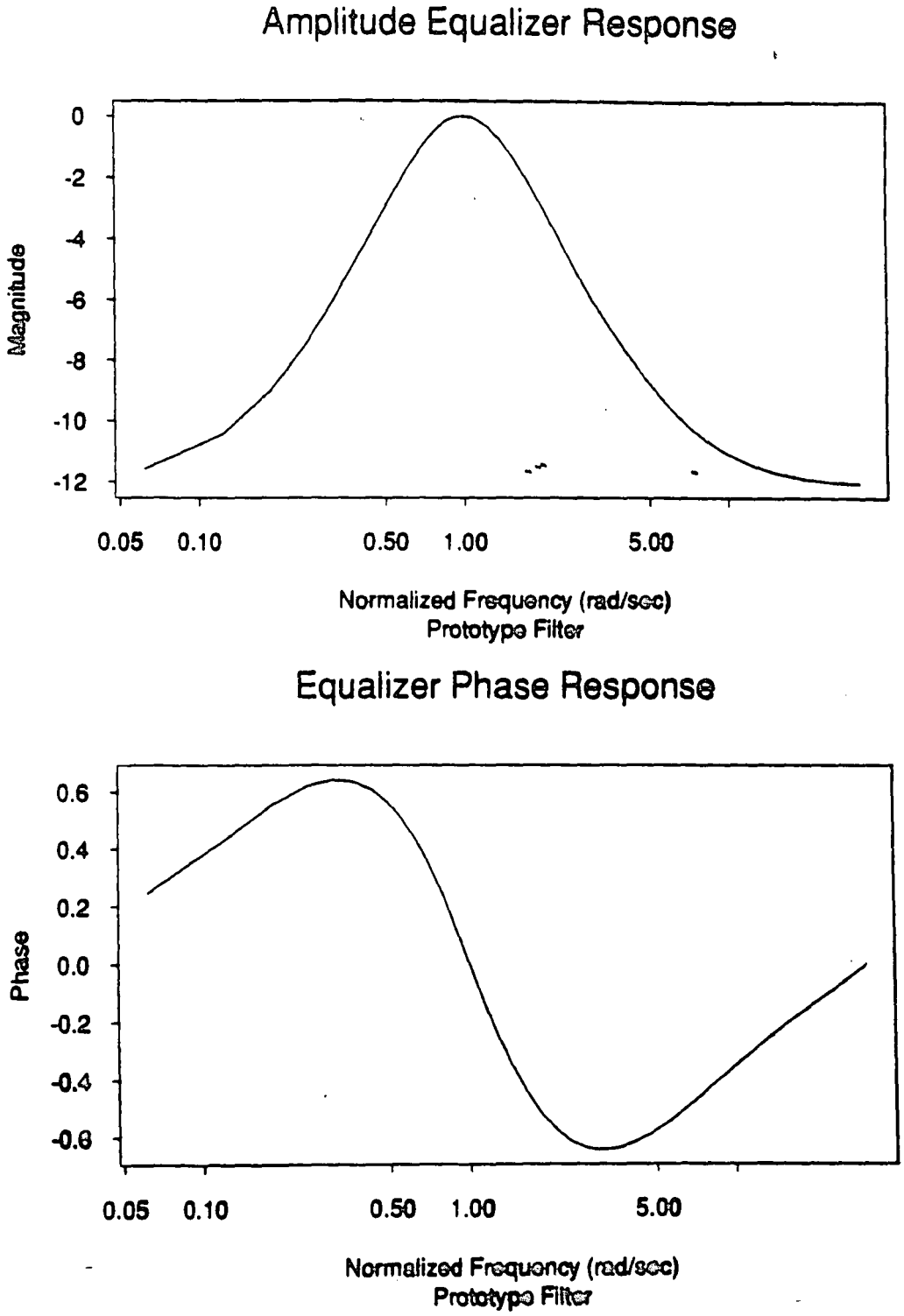


FIGURE 18. Input and Desired Signals

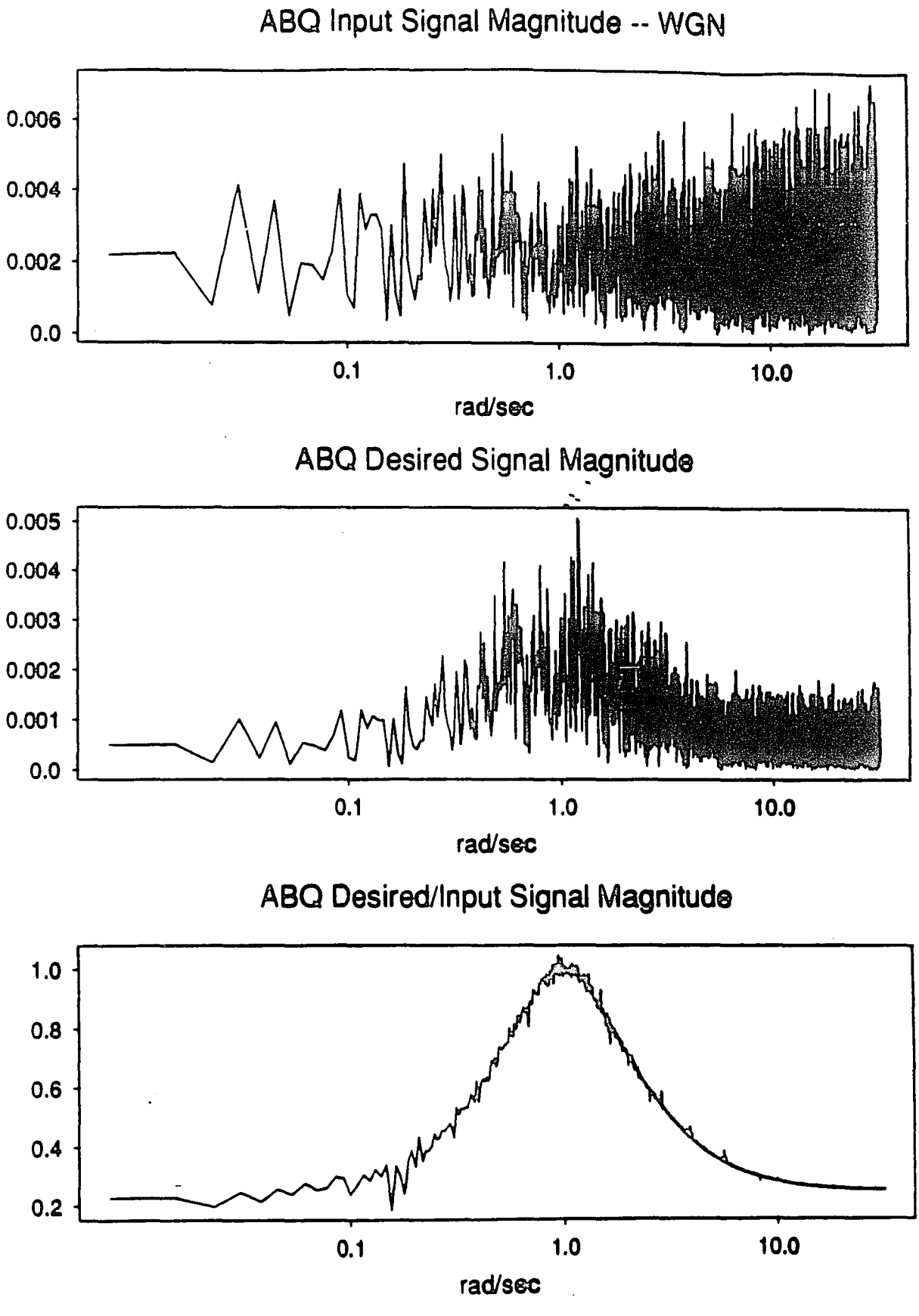


TABLE 2. Amplitude Equalizer Model Gradient Filters

	A0	A1	A2
Grad0	0.9322	-1.8645	0.9322
Grad1	0.046614	0	-0.046614
Grad2	0.002331	0.004661	0.002331

The LMS gain was chosen empirically. An arbitrary starting gain of 0.1 was chosen and the behavior of the MSE was observed for 8K iterations. The LMS gain was decreased until the MSE exhibited an exponentially decreasing trend. For this experiment an LMS gain of 0.03125 was found to give good results. Table 3 shows the filter behavior at 8K intervals from the initial conditions to the adapted conditions:

TABLE 3. ABQ5B Amplitude Equalizer Behavior LMS Gain=0.03125 WGN input

Iter	A0	A1	A2	Z1	Z2
0	1	-0.6	0.05	0.5	0.1
8K	0.5294	-0.5884	-0.0321	1.1634	-0.05214
16K	0.2820	-0.45534	0.1826	0.8723	0.7433
24K	0.3003	-0.4057	0.1671	0.9864	0.5641
32K	0.2989	-0.4649	0.1683	0.9813	0.5738
Desired	0.2990	-0.4649	0.1682	0.9817	0.5730

The performance of this filter is shown graphically in Figures 19 and 20. Figure 19 shows the trajectories of the simulator filter coefficients over the first 16K iterations. Figure 20 shows the learning curve, the magnitude and log magnitude of the mean square output error, for the entire 32K iterations. The MSE was calculated with a 10 sample wide running average FIR filter. In Figure 19 we see that the coefficients exhibit ringing, or damped sinusoidal behavior. From this we conclude that our system is underdamped. The final values of the zeros are less than 0.1% away from the desired values. In Figure 20 we see that the MSE seems to be decreasing as a nearly constant exponential rate. From these observations we can conclude that out system is working and that it is working very well.

FIGURE 19. Coefficient Trajectories Zero Only Adaptation

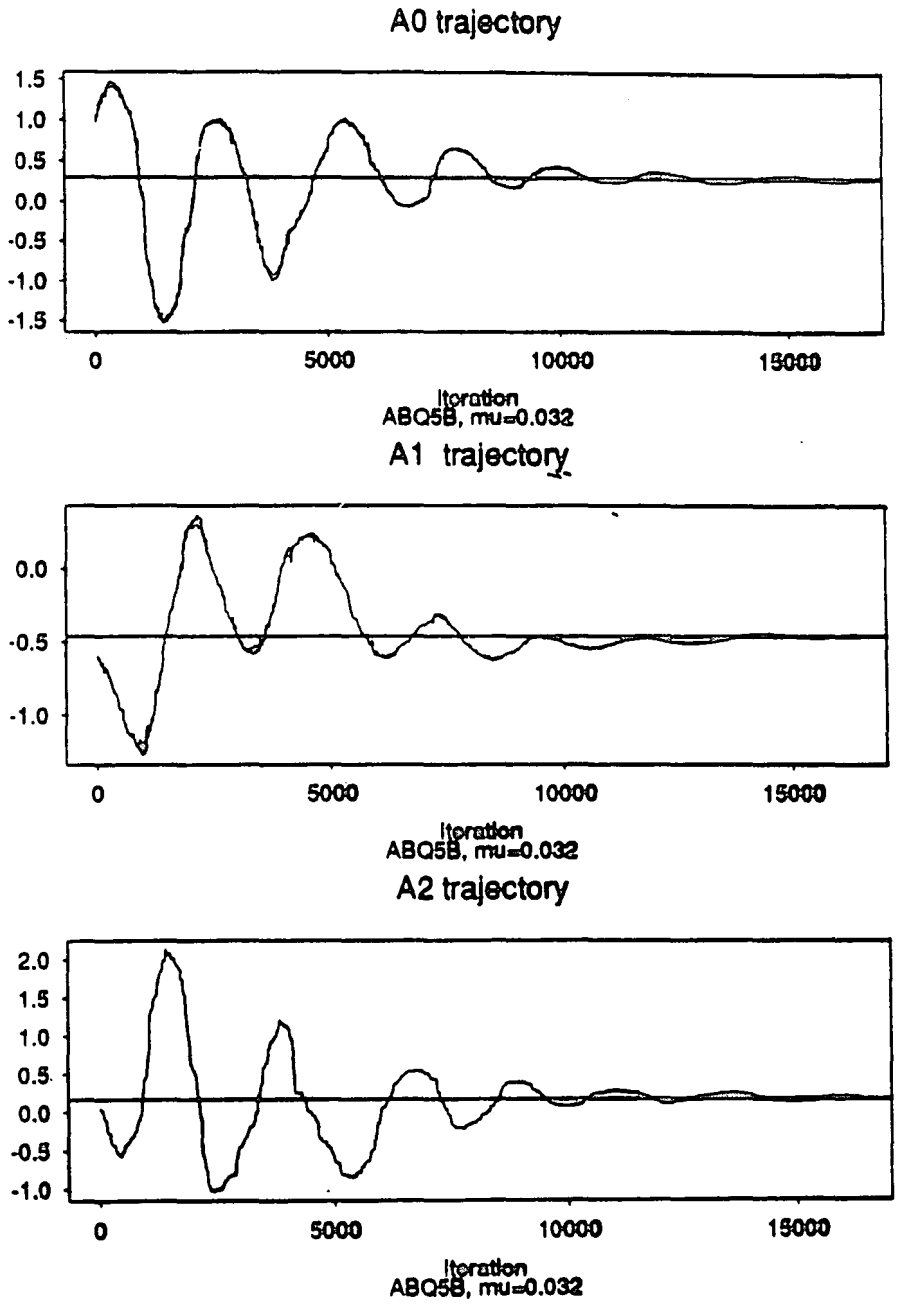
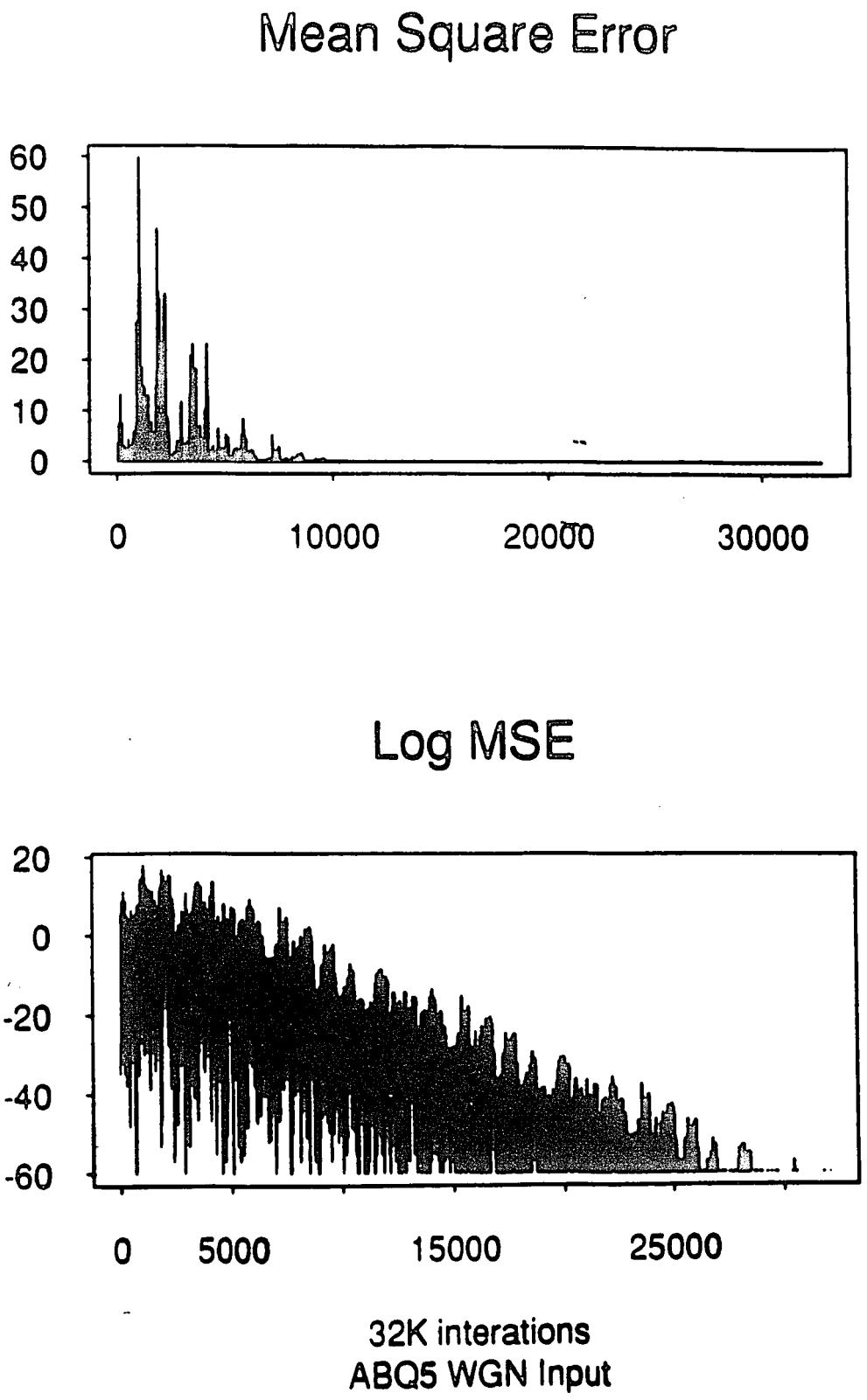


FIGURE 20. Learning Curves Zero Only Adaptation



There are some unsettling issues. The under-damped behavior of the coefficients is not desirable. We would prefer to have the LMS algorithm system poles so far from the imaginary axis that variations in input power or initial conditions can never make the system unstable. Since we know that the stability of our system will depend to a large extent on the spectral characteristics of the input signal and on the LMS gain [10]; the next experiment explores the stability with variations in the LMS gain

5.1.2 Variations in LMS gain

For the exact same system as we used before, we varied the LMS gain and observe the behavior of the adaptive coefficients and the mean square error. We used the same initial conditions as for the first experiment but looked only at 8K iterations. Stability was judged with two criteria.

1. Are the coefficients behaving in an overdamped, underdamped or oscillatory manner?
2. Is the MSE continually decreasing in an exponential manner?

TABLE 4. Model Matching Stability

LMS Gain	Coefficient Behavior	MSE Behavior
0.1	oscillatory	increasing
0.05	underdamped	decreasing
0.025	underdamped	decreasing
0.0125	underdamped	decreasing
0.00625	underdamped	decreasing

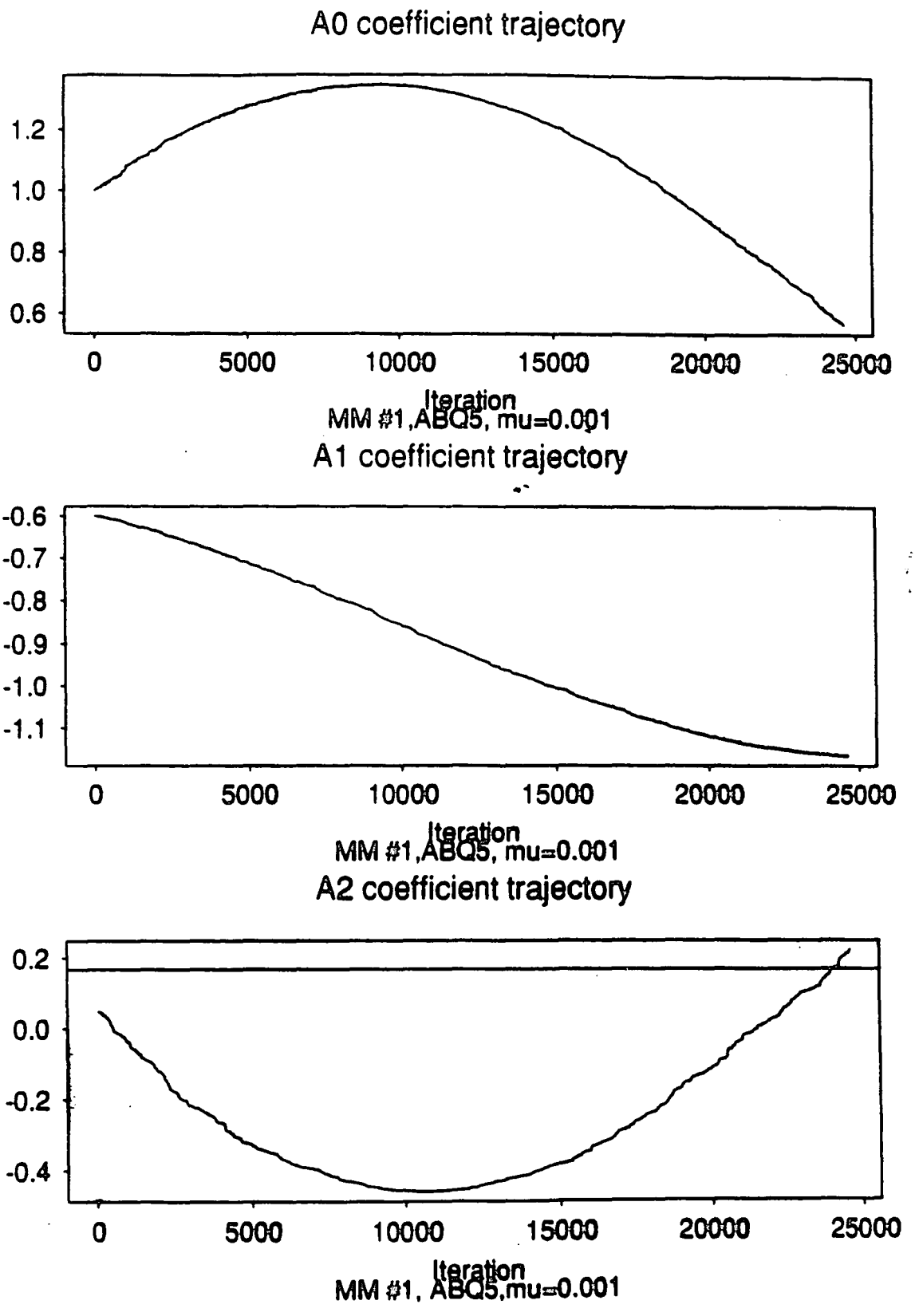
Based only on these observations it is difficult to draw any conclusions. For large values of LMS gains the system is clearly unstable. As the LMS gain is decreased the period of the oscillations in the adaptation algorithm increases. This means that systems with very low values of LMS gain observed for short period of time may appear overdamped while in fact the period of oscillation is simply much longer than the observation time. The implications of this are that it is always necessary to ensure that simulations are carried out long enough to cover several LMS adaptation periods and that

it may be desirable to monitor the error during adaptation and adjust the LMS gain based on the rate of change of the error. This makes a strong case for extensive system simulation before making a commitment to hardware.

It is useful to look at the behavior of an underdamped algorithm with a long time constant. Figure 21 shows the trajectories of the same filter with an LMS gain of 0.001.The system is still underdamped. We can assume that eventually it will converge on the optimal solution just as it did with an LMS gain 31 times larger, and we expect, a time constant 31 times longer. One could be fooled by looking over a shorter time interval into believing that the system was over damped. It is to be expected that eventually by reducing the LMS gain one could have a system with a clearly overdamped response. The increase in adaptation time may or may not be warranted.

What have we found so far? We have found that we can simulate a continuous time filter with adaptive zeros with a discrete time model. We have found that our derivation of an LMS adaptive algorithm works and that, as we expected, the LMS gain controls both the stability and the rate of convergence.We have found that this system tends to be underdamped even for very low values of LMS gain and that if we insist on being overdamped then we will pay a tremendous penalty in convergence and simulation time. We found that we may wish to design such a system with an adaptive LMS gain where the adaptation is based on the rate of change of the error. We have also demonstrated the need for extensive system level simulations before commiting a design to hardware. We next wish to examine the behavior of the filter for a fixed LMS gain as we change the input power and or the type of input signal.

FIGURE 21. Coefficient Trajectories with Small LMS Gain



5.1.3 Adaptation with Signal Power Variations

We can study filter behavior with variations in signal power by scaling the output of the noise generator program.We continue to work with white gaussian noise (WGN)and scale both the input signal and the desired signal by the same scale factor. Both signals are scaled so that the filter is being presented with the same problem. When we scaled by a factor of two we found that the algorithm was no longer stable for the same value of LMS gain. After a number of experiments we concluded that for a given signal power there is a maximum value of LMS gain for which the algorithm is stable. We also found that the relation given by Widrow and others [35], $\mu < \frac{1}{(L+1)P}$, where L is the number of taps and P is the input signal power, did not work for this filter. There are three possible ways of dealing with this problem:

1. Choose a value of LMS gain such that stability is guaranteed for the largest possible input power and accept the long convergence time for lower power.
2. Provide an AGC stage such that the input power is fixed to a known limited range.
3. Use a variable LMS gain that is linked to the input signal.

Methods 2 and 3 are essentially the same. The algorithm is sensitive to the input power both through the signal path filter and the error but also through the gradient filter outputs. While not explored in this work, the normalized gradient LMS algorithm may reduce this sensitivity[21]. This algorithm normalizes the output signals of the gradient filters and removes any sensitivity to input power variations. Since it is probably necessary to AGC the input signal in any case, this algorithm may be redundant for practical applications.

5.1.4 Convergence with non-zero mean signal

In general, one cannot expect a stochastic algorithm to converge with a non-zero mean signal. However, it is quite likely that due to DC offsets, a continuous time adaptive filter will have a non-zero mean signal. To investigate the sensitivity to DC we maintained

the zero mean desired signal and introduced small offsets into the input WGN. We let the filter adapt for 8K iterations and observed whether or not the algorithm converged. Table 5 shows the filter behavior as small offsets are added the input signal.

TABLE 5. Model Matching Convergence with Non-Zero Mean Signal and LMS gain = 0.03125

Mean input	s.d	Stable?	
0.001	0.2287	yes	standard input signal
0.005	0.2287	yes	
0.010	0.2287	yes	
0.050	0.2287	yes	
0.150	0.2887	yes	
0.200	0.2887	marginal	
0.250	0.2887	no	

With a dc offset of 0.15, the peak overshoot of the coefficients was larger than for the zero mean case but the trajectories were still well behaved damped sinusoids. With a dc offset of 0.200, the coefficients start to exhibit strange, noisy, behavior and at a dc offset of 0.25 the filter is completely unstable. These results are encouraging in that they imply that we need not take more than ordinary care in minimizing dc offsets in our filter.

5.1.5 Operation with Multi-tone Signal

White noise is not the only input signal that an adaptive filter must deal with. While in most communications systems scrambling and coding are used to whiten the transmitted signa, there must always be strong components at the clock frequency to provide sufficient information for timing recovery. There are also many non-communications applications for adaptive filters. It is important to investigate the behavior of this filter with correlated input signals. We first attempt to adapt with a multi-tone signal.A signal composed of 10 uncorrelated tones of approximately equal power was created. We then shaped this signal with the prototype amplitude equalizer and used this output signal as the desired signal for the adaptive filter. No noise was added to either the input or the desired signal.We let the filter adapt for 8K iterations with an LMS gain of 0.03125 and found the results to be quite encouraging. Figure 22 shows the frequency

spectrum of the input signal, the desired signal and the ratio of the input and desired signals. Figure 23 shows the coefficient trajectories and Figure 24 shows the movement of the zeros. The convergence transient response for the multi-tone input signal is quite different form the response for the WGN input. The "fuzziness" of the trajectories in Figure 23 is associated with the high frequency behavior of the error estimate. This could be reduced by decreasing the LMS gain. Experience with observing the coefficient trajectories indicates that the fuzziness may tend to indicate near borderline instability. This implies that we may need to reduce the LMS gain for correlated inputs. The behavior of the zeros, shown in Figure 24 is very reasonable. They are very close to the optimal solution within 4K iterations.

The change in the transient properties of the algorithm may be explained by the change in the correlation in the input signal. It is known that, for the FIR filter case, the gain of the feedback path is determined by both the LMS gain and the power spectral density, or autocorrelation function, of the input signal. The input signal statistics as well as the absolute power affect the stability and convergence of the algorithm[10]. It is interesting to examine the autocorrelation functions of the multi-tone input and the gaussian noise inputs.For the WGN input there is no correlation between input samples. For the second case, even though the input tones are not harmonically related, there should be more correlation between samples. It is known that when the input autocorrelation matrix is ill-conditioned, that is the eigenvalues are widely spread, then the settling time will tend to be dominated by the slowest mode, i.e. the smallest eigenvalue.

FIGURE 22. Multi-tone Input and Desired Signals

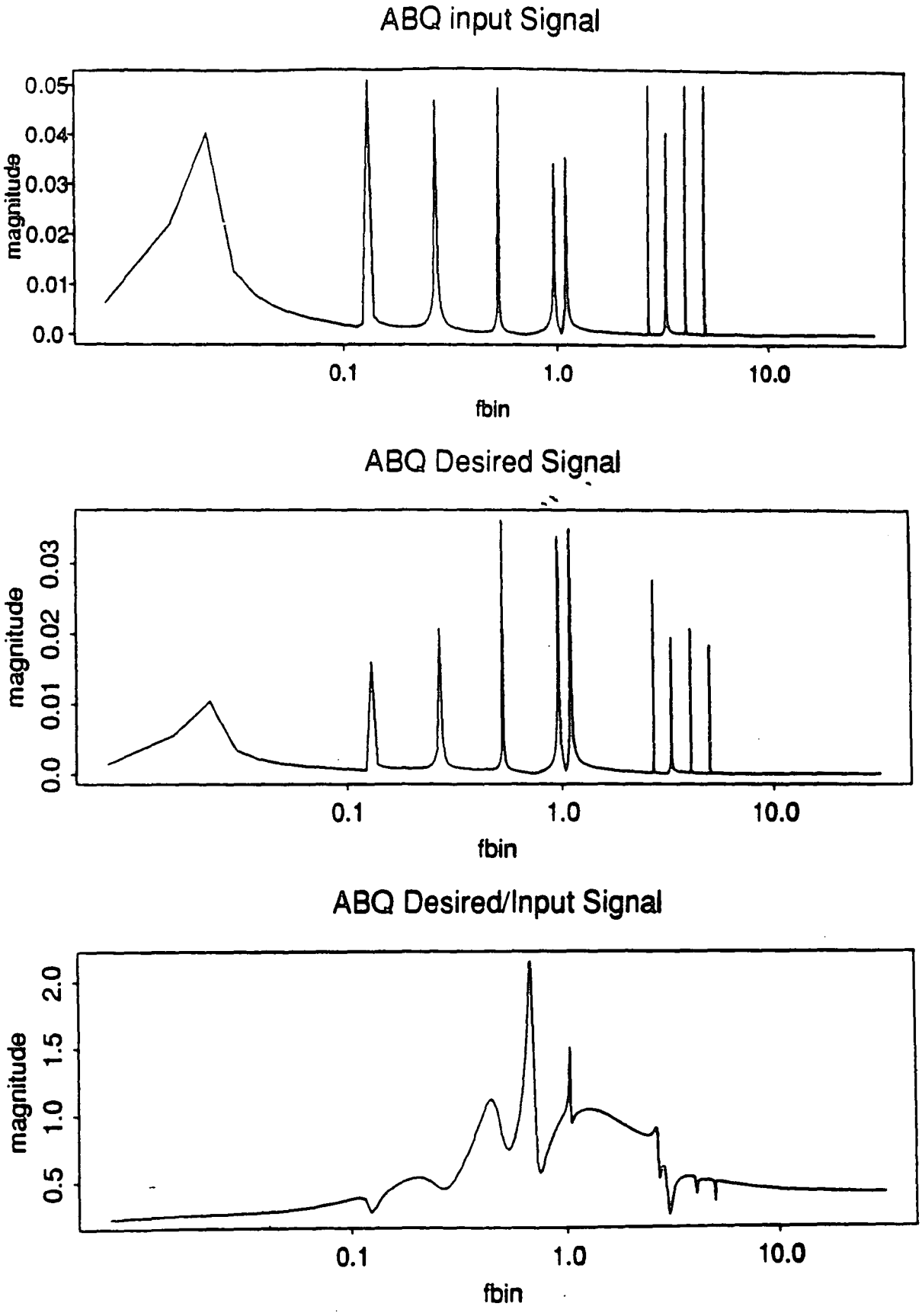


FIGURE 23. Multi-tone Coefficient Trajectory

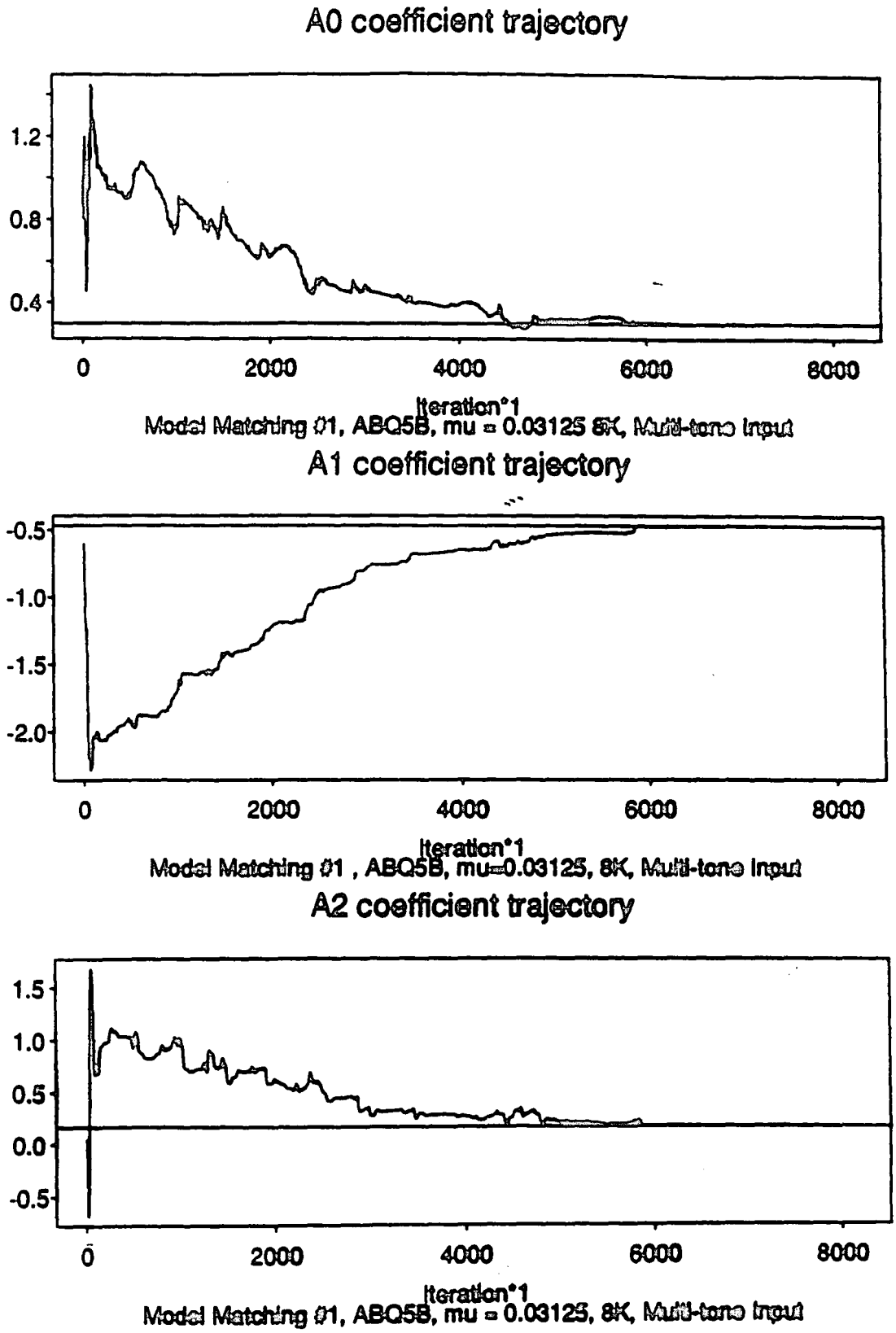
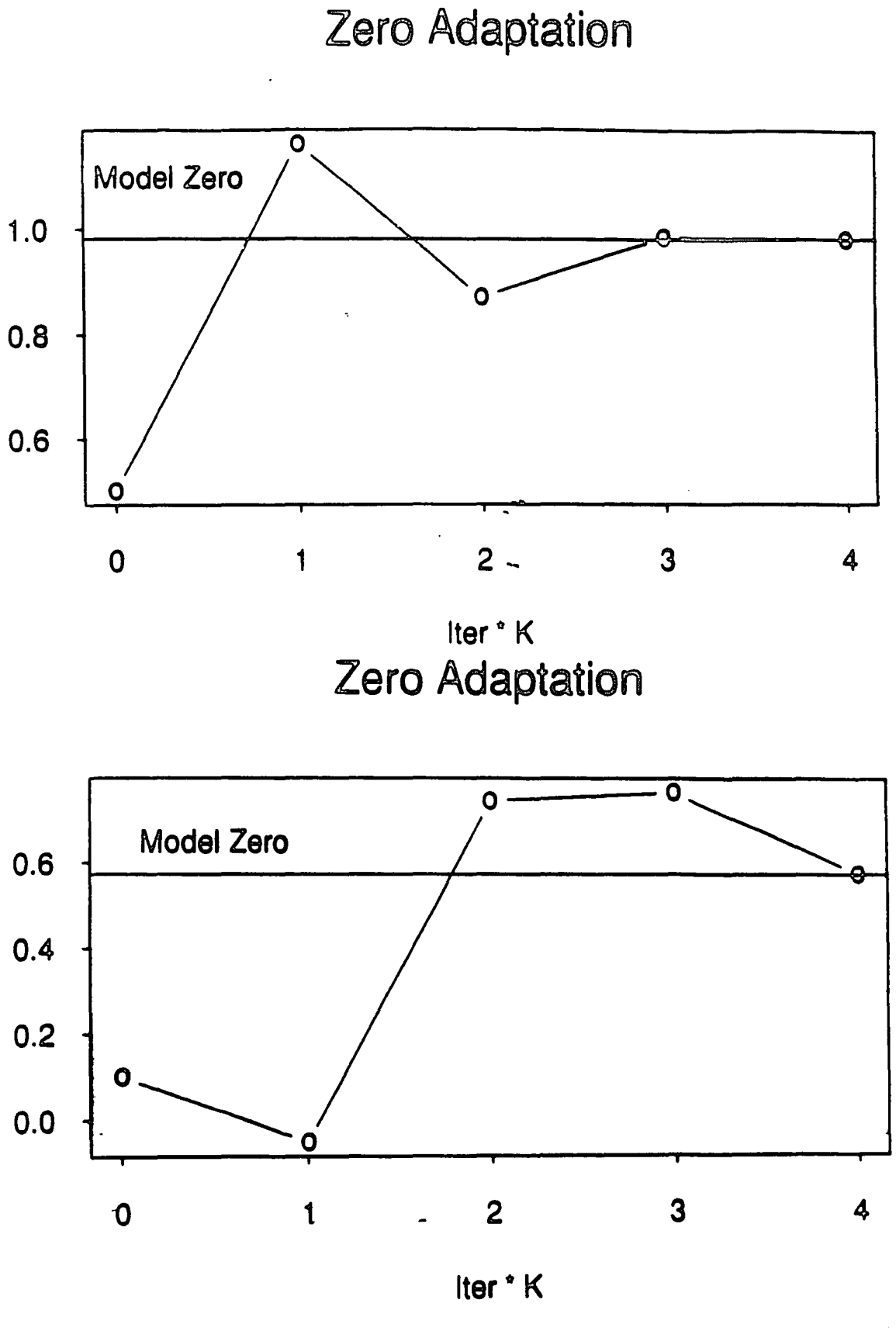


FIGURE 24. Zero Movement - - Multitone Input Signal



If we form the autocorrelation matrix and solve for the eigenvalues for the two cases we get

TABLE 6. Multi-tone Autocorrelation Matrix:

1.0000000	0.9685555	0.8818408	0.7541434
0.9685555	1.0000000	0.9685555	0.8818408
0.8818408	0.9685555	1.0000000	0.9685555
.7541434	0.8818408	0.9685555	1.0000000

The eigenvalues are:

3.7148	0.27653	0.00785	0.00076

TABLE 7. Autocorrelation Matrix for Gaussian Noise

1.0000000000 -	0.0003166881	-0.0009870712	-0.0023286797
-0.0003166881	1.0000000000	-0.0003166881	-0.0009870712
-0.0009870712	-0.0003166881	1.0000000000	-0.0003166881
-0.0023286797	-0.0009870712	-0.0003166881	1.0000000000

The corresponding eigenvectors are:

1.0025	1.0003	1.0001	0.9970

The eigenvalues for the multi-tone source are well separated while the eigenvalues for the WGN noise source are tightly clustered. Since the transient behavior of the LMS algorithm is roughly inversely proportional to the sum of the time constants for each natural mode, where with natural modes are set by the product of the eigenvalues and the LMS gain, [10] then for the tightly packed case we conclude that there is a high degree of interaction between the modes and the transient response exhibits the kind of characteristics we associate with complex multi-mode systems. In the case where one eigenvalue is clearly dominant, then that mode controls the transient response. This is analogous to the dominant pole compensation used in control systems.

The fact that the natural modes of our adaptive algorithm are so tightly coupled to the statistics of the input signal is somewhat disheartening. The implication is that we either must know the statistics or at least the bounds of the statistics or else try to dominate the system with the LMS gain. As we have seen, this can require very small values of gain.

5.1.6 Error Impulse Response

Another way of studying the loop dynamics of an adaptive filter is by studying its response to an impulse in the error signal. This is similar to what may happen when after the filter has been operating for a long time with one set of input conditions there is a sudden change in the environment. We can simulate this situation by:

1. Using the system denticulation arrangement we initialize the adaptive filter coefficients to the optimal values, 1.e. the known values of the prototype filter.
2. We operate the filter in the normal manner. The output error is 0.
3. We introduce an impulse into the desired response signal. This causes an impulse in the error signal.

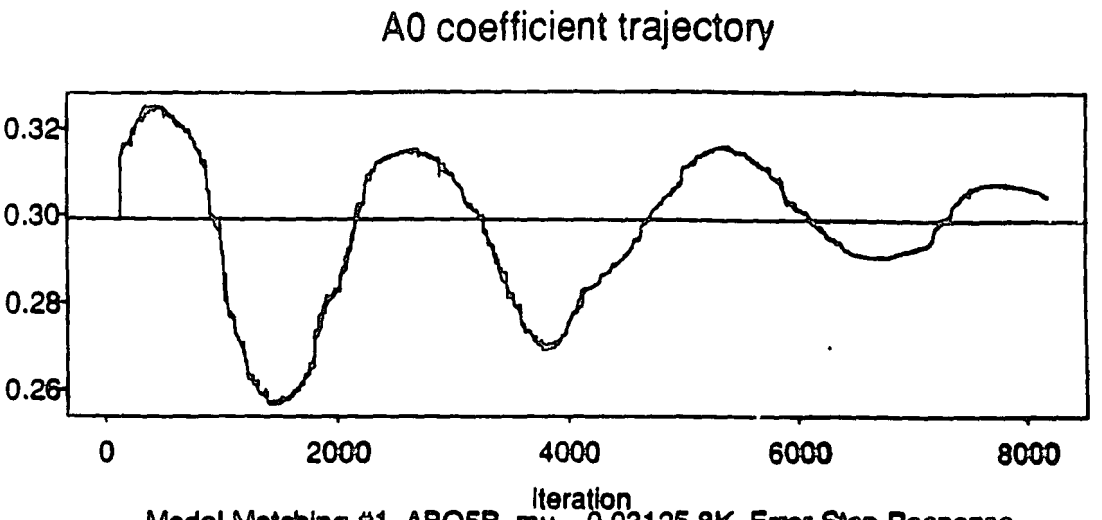
We can then observe the behavior of the coefficients as a result of the error impulse. When we perform this experiment we observe the decaying sinusoidal response of the coefficients. We can find the normalized frequency by taking the FFT of the difference between the desired coefficient and the actual coefficient. The poles of the LMS algorithm are small compared to the resolution of the FFT.

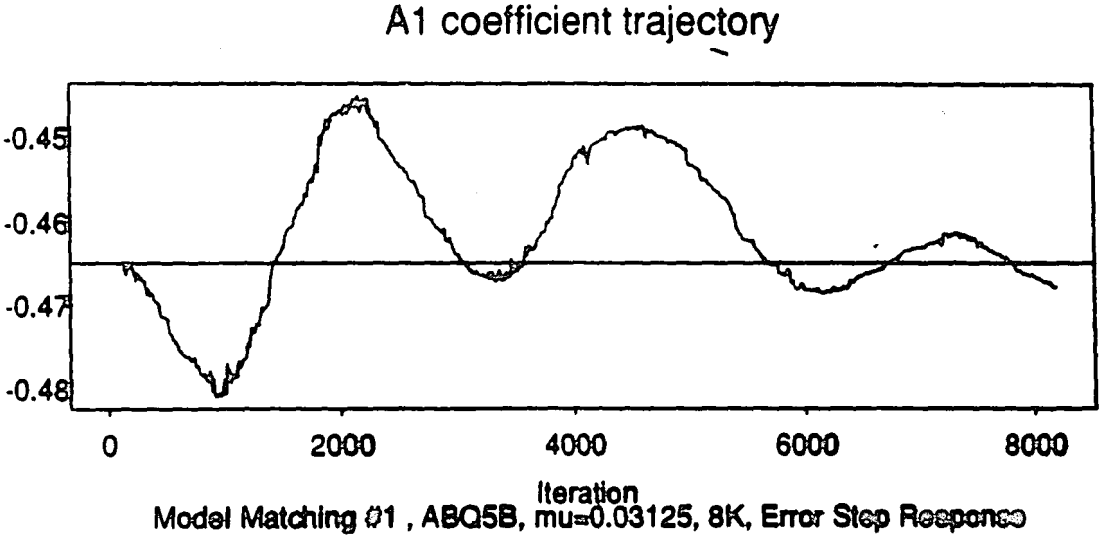
TABLE 8. Measured Natural Modes of Coefficient A Error Impulse Response

LMS Gain	FFT bin	normalized frequency Rad/Sec
0.0625	7	0.00854
0.03125	3	0.00366
0.015625	1	0.00122

As we scale the LMS gain by a factor of two the frequency also scales by about a factor of two. What the table doesn't show is that as we increase the LMS gain the behavior becomes less of a pure damped sinusoid and more complicated. Figures 25 and 26 show the error step response for two cases of LMS gain. Figure 26 clearly shows the complicated behavior of the system. The choppy behavior shown in Figure 26 is indicative of a system near the edge of instability. As with any control system, the impulse response can be used to study the stability of the system.

FIGURE 25. Error Impulse Response with LMS Gain = 0.03125





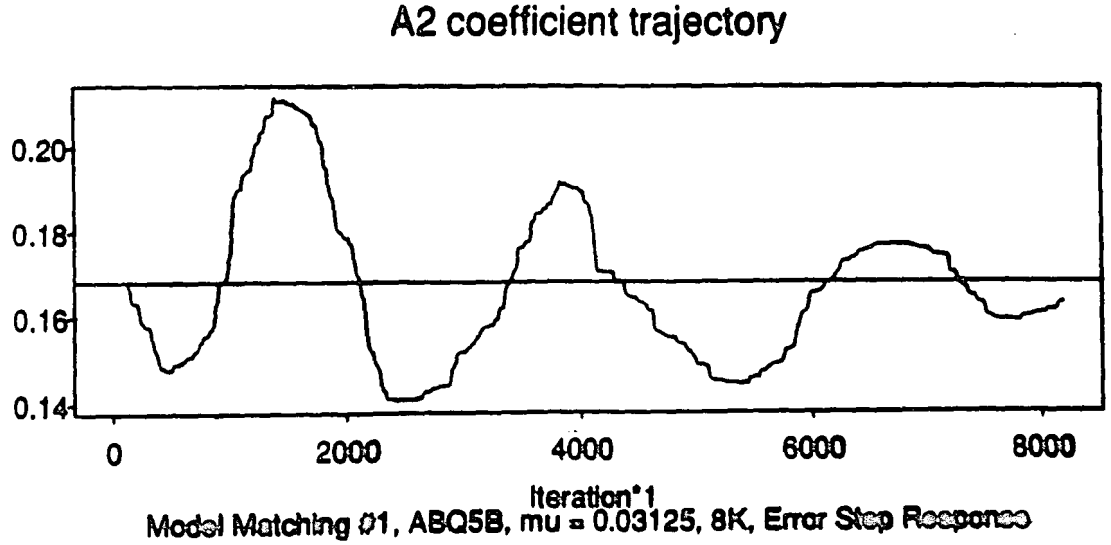
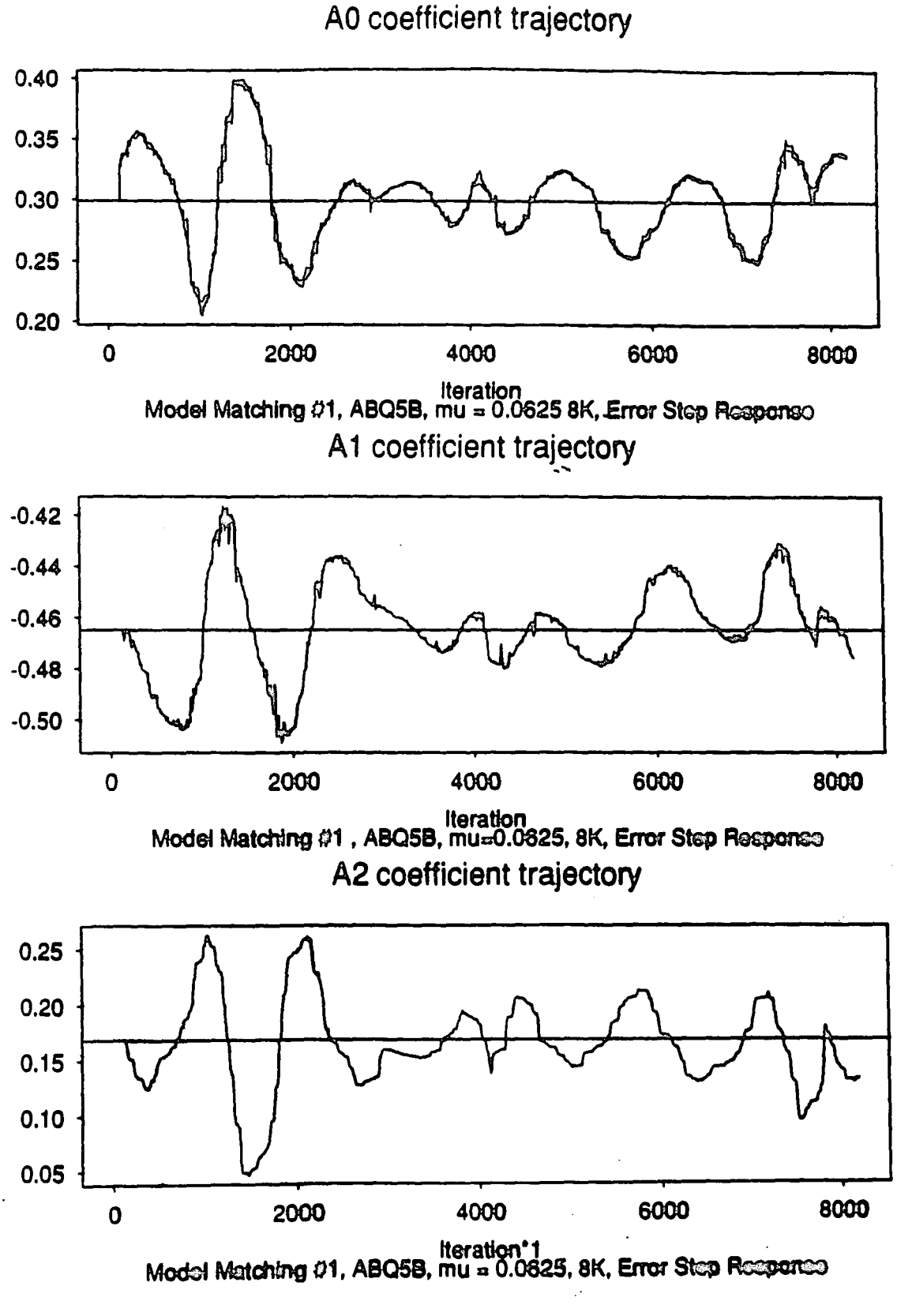


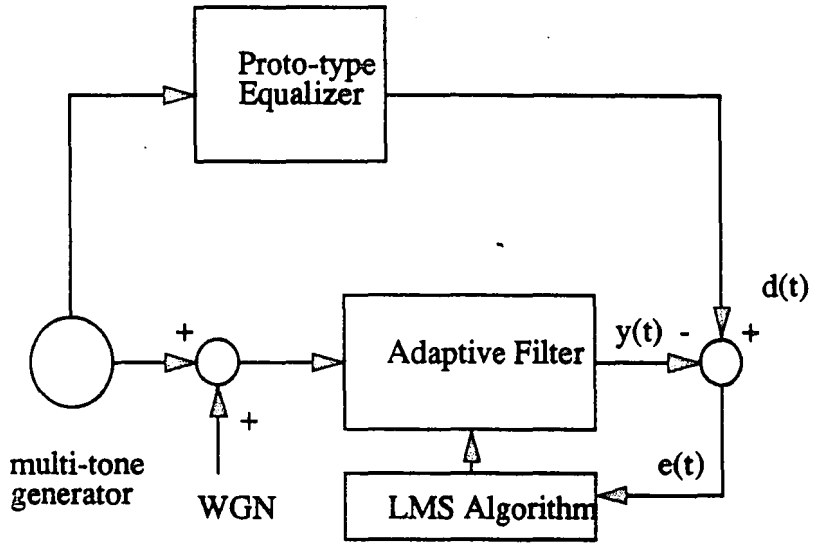
FIGURE 26. Error Impulse Response with LMS Gain = 0.0625



5.1.7 Operation with Variations in Input Signal to Noise

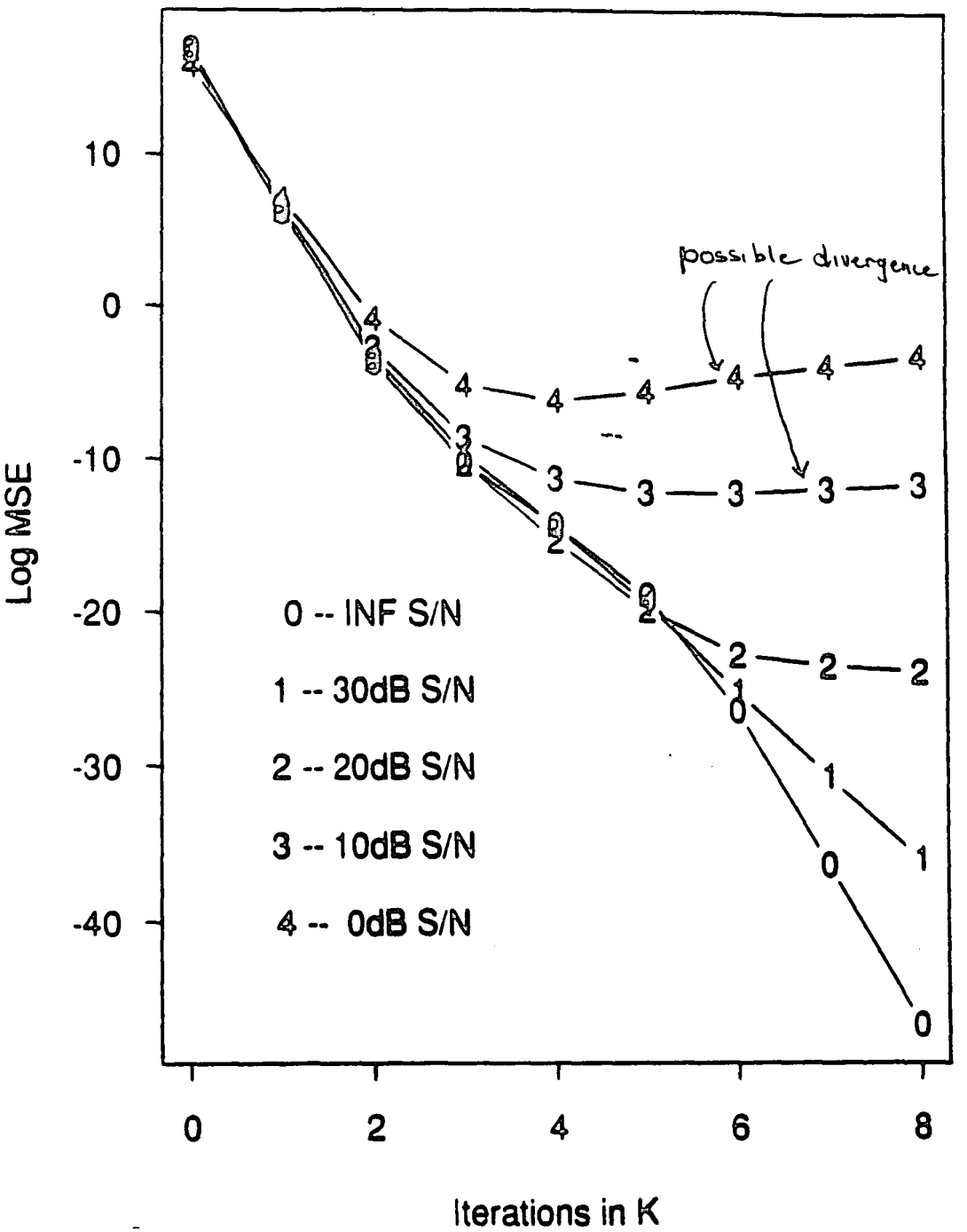
We can use the multi-tone input signal to experiment with variations in the input signal to noise ratio. For this experiment we shape the pure multi-tone signal with the prototype filter and use that signal as the desired signal input to the adaptive system. We then add various amounts of WGN to the multitone before applying it as the input to our filter. Figure 27 shows the configuration.

FIGURE 27. Signal to Noise Experiment



The input signal-to-noise ration was varied while the input signal power was held constant. The filter adapted for 8K iterations and the MSE was calculated and plotted every 1K iterations. Figure 28 shows the results. There is excellent MSE reduction with 20dB S/N. At 10 and 0 dB S/N there is less reduction although the performance might be much better if the system was allowed to adapt for much longer periods. There may be some divergence of the MSE curve for lower S/N ratios. This may be caused by rippling of the output error due to the underdamped nature of the system. While more study is needed, these results indicate that the adaptive zeros filter will be able to perform in a wide variety of applications even with fairly low Signal to Noise ratios.

FIGURE 28. Learning Curve for Various S/N Ratios



5.1.8 Operation with Two Uncorrelated Tones

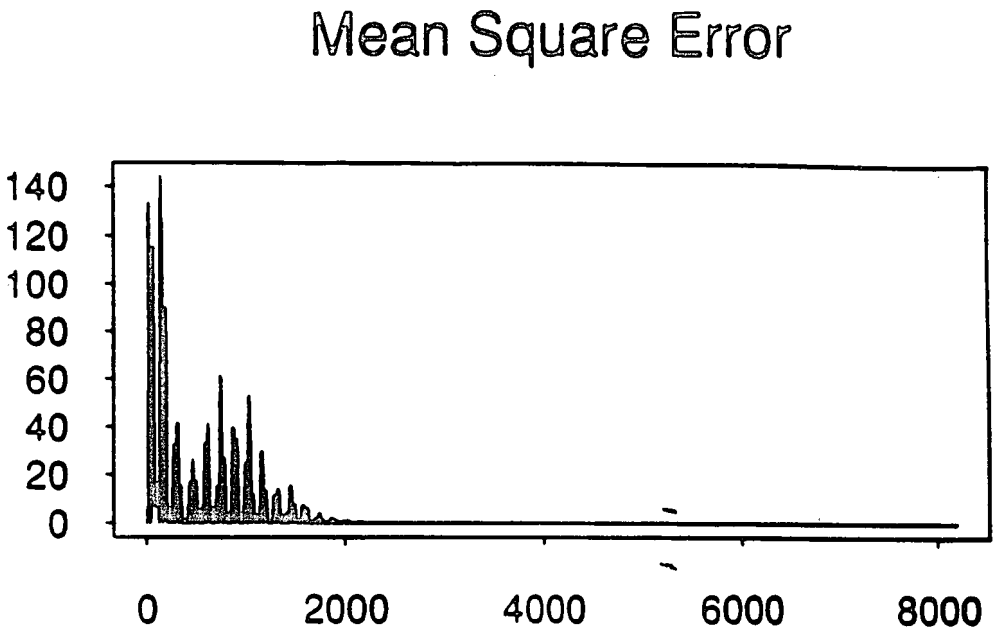
One major application for adaptive filters is interference cancelling. In this application it is necessary to operate the filter not with noise or noise-line inputs, but with a small number of tones and possibly very low noise levels. An experiment was set up to determine if the filter would converge in a model matching application with only two tones. Obviously we cannot expect to tune the filter to match the prototype filter over the entire band with information at only two input tones. A signal was created with tones at 0.97 and 2.7 rad/s. White noise was added to the signal and the prototype filter was used to shape the resultant signal. The adaptive filter was then run using the output of the prototype as the desired signal. After 8K iterations the coefficients had converged to:

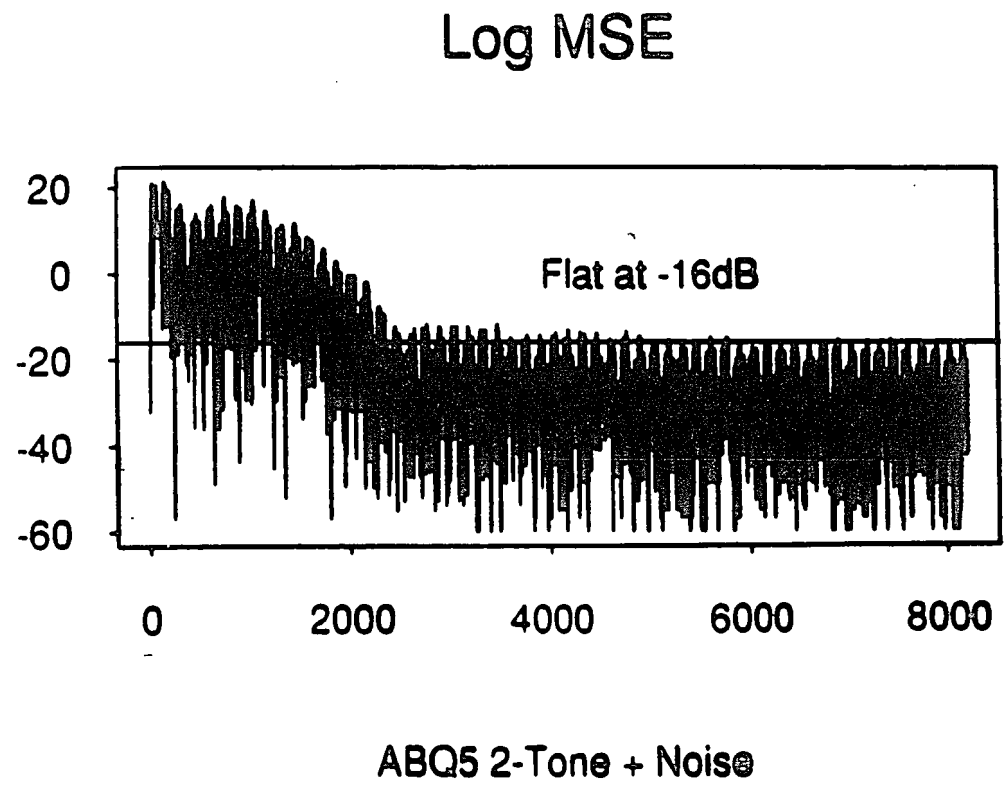
TABLE 9. Double Tone Convergence

	A0	A1	A2	Z1	Z2
Initial	1	-0.6	0.05	0.1	0.5
Des	0.2989	-0.4649	0.1683	0.9817	0.5730
Final	0.5801	-1.0905	0.5168	0.9398	+j0.0826
				0.9398	-j0.0826

with an LMS gain of 0.001. While it looks as if the filter has not done a very good job, the mean MSE over the last 1K iterations is less than 0.028 (-15dB). What has happened here is that the filter has converged on the optimal solution based on the information present in the desired and input signals.The strong tones dominate the adaptation process. During the course of this experiment it was noted that one must be very careful when attempting operation with only a few tones. In general a single tone may not provide enough information for the LMS algorithm to converge. In a a two-tone application if one tone is in the filter's stopband it may be difficult or impossible to achieve convergence. The addition of another inband tone or more noise power may help in this situation.

FIGURE 29. Learning Curve for Two Tone Input



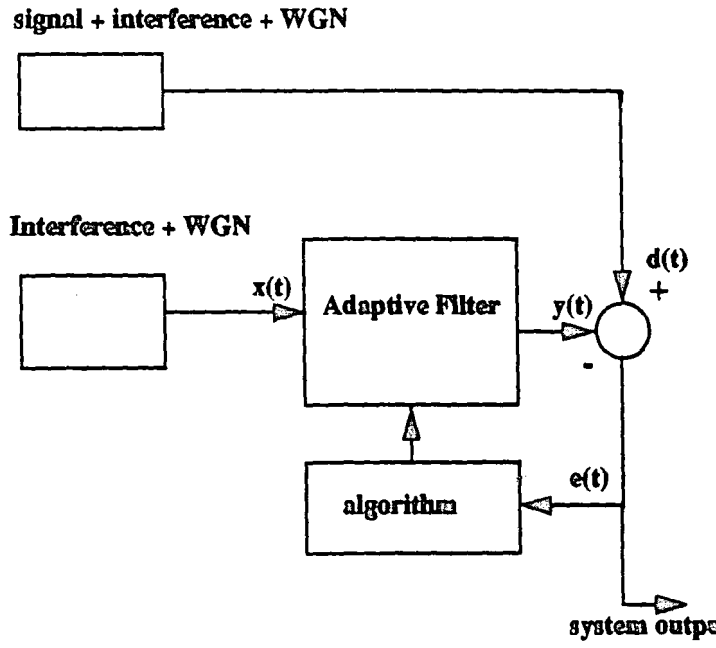


In this case the final filter response is wrong as far as the systems identification problem. However, as shown by the learning curve, the filter is doing the best it can from a mean square error perspective. We see that the MSE flattens out at about -16dB. This demonstrates the importance of designing the characteristics of the desired signal such that the filter has sufficient information the properly adapt.

5.1.9 Interference Cancelling Experiment

In order to test this filter in a practical application, an interference cancelling experiment was devised as shown in Figure 30. This is a classic experiment presented by Widrow in [35]. The adaptive filter looks at the input interference tone and the input signal and adapts the output of the filter such that the total output noise is minimized.

FIGURE 30. Interference Cancelling



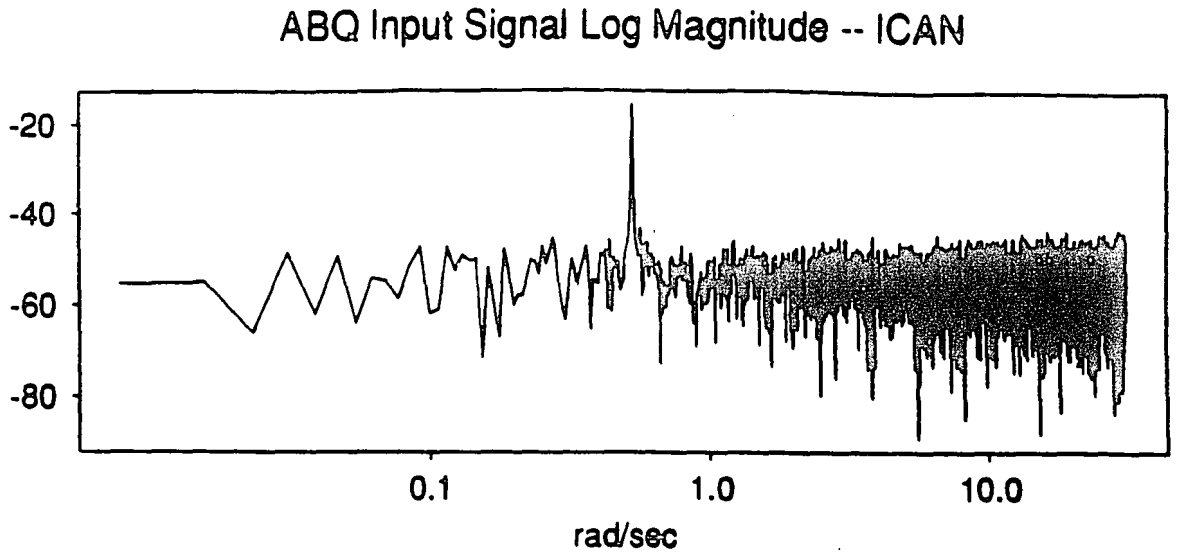
The result is that the adaptive filter will try and notch the interference tone from the desired signal. In this case the error signal is the system output. Figure 31 shows the results of this experiment. The top plot is the input, X(s), to the filter. This consists of the low frequency interference and WGN. The middle plot is the desired signal consisting of

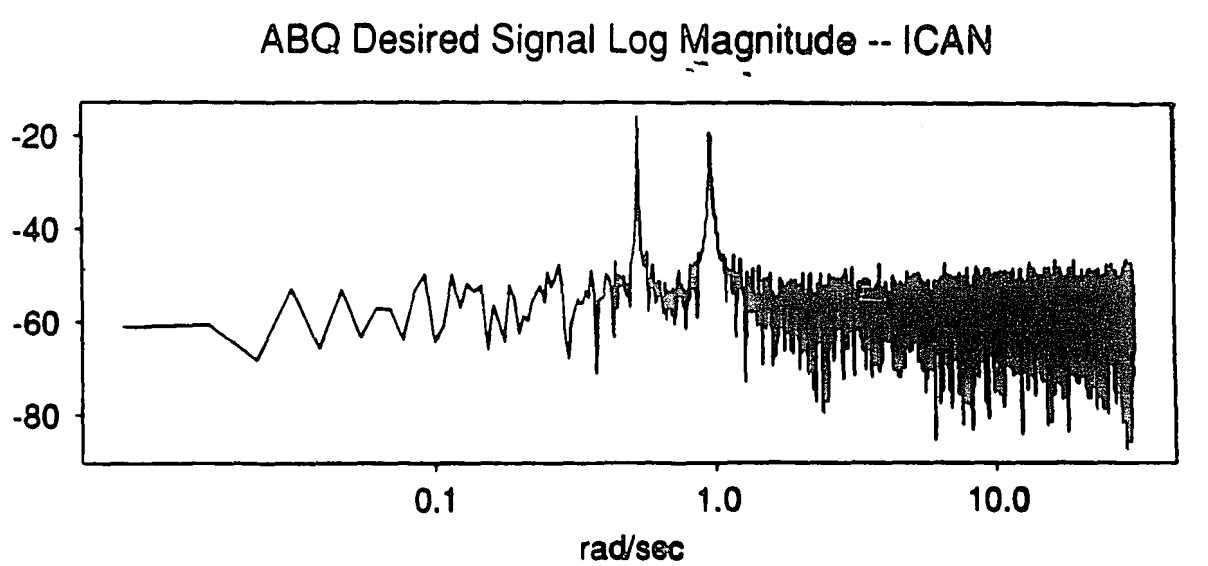
the desired tone, the interference tone and WGN. The lower plot is the output error during the adaptation process. We see that the interference tone is quite effectively notched out of the signal. The amplitude of the error is small but flat gain will easily take care of that. The interesting low frequency tone and the fuzziness of the signal tone are an artifact of the adaptation process. These show the output with a non-stationary transfer function. In a real application one would probably let the system adapt for a while and then stop the process to eliminate the problems with a non-stationary transfer function. Figure 32 shows the initial and final transfer functions of the filter. The filter has adapted itself from a low pass filter with large gain to a non-symmetrical bandpass filter with a peak near the desired signal and attenuation near the interference tone.

5.1.10 Summary

This concludes the experiments on the adaptive zero, fixed pole filter. This filter has been found to be practical and to require very little extra hardware. It was found that the selection of the LMS gain is best done through extensive system simulation and that it may be desirable to use a time-variable gain. The use of automatic gain control to fix the input power to the filter was found to be desirable in practice. It was also found that the initial locations of the zeros did not have a major impact on the stability or convergence of the filter. The filter was found to function with small dc offsets and with fairly low input Signal-to-Noise ratios. It is likely that many potential applications of continuous time adaptive filters may be well served by the simplicity of the adaptive zero, fixed pole adaptive filter.

FIGURE 31. Interference Cancelling Experiment Results





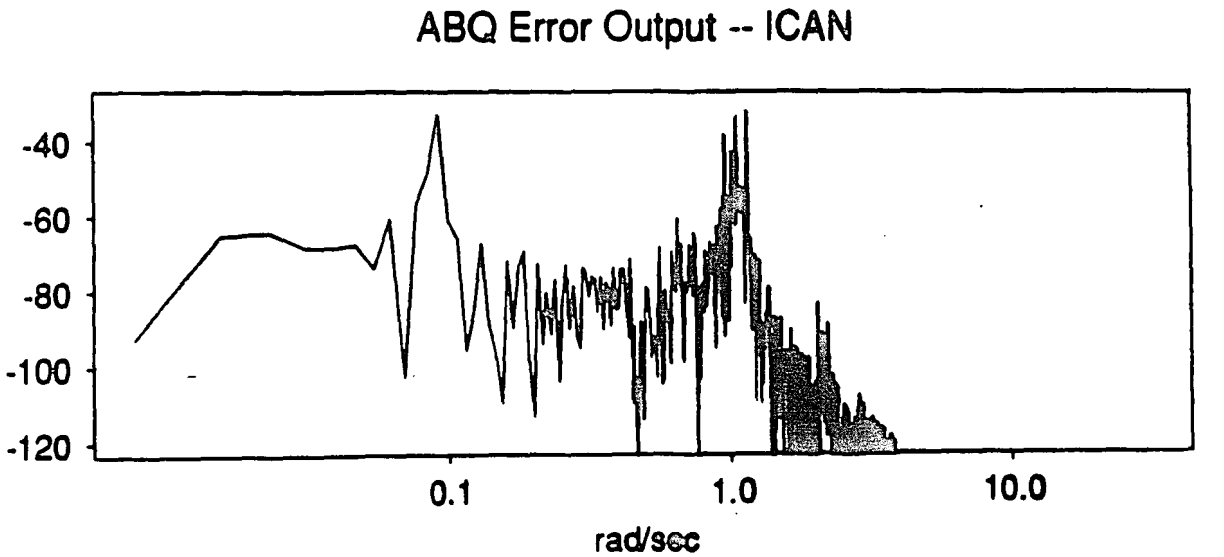
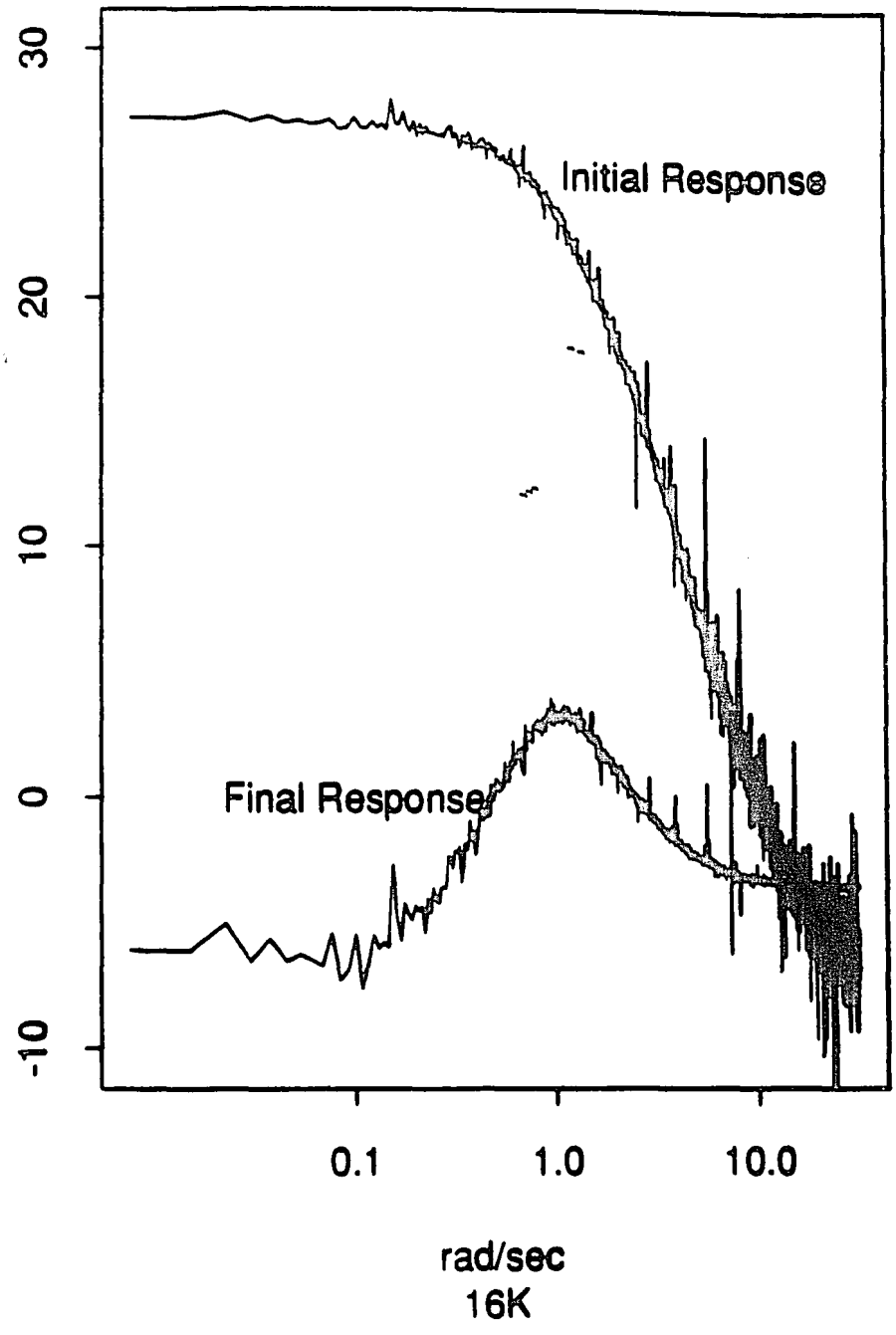


FIGURE 32. Interference Cancelling Experiment Transfer Function



5.2 Output Error Experiments

5.2.1 System Identification with WGN Input

The basic experiment for the output error formulation of the system identification problem was the same as for the zero only filter case. The prototype equalizer was defined as the system to be identified, the input signal was WGN and the desired signal was WGN shaped by the prototype filter. The initial zero locations were chosen to be the same that were used in the zero only case, the choice of initial poles was not nearly so arbitrary. The gradient filters were initialized with the same initial poles as the signal filter. The filter parameters are shown in Table10.

TABLE 10. Output Error Experiment Filter Parameters

	Continuous Time	Discrete Time
Zeros	z1 = -0.184238	0.5731
	z2 = -5.4277	0.9817
Poles	p1 = -0.7015 +j 0.7127	0.92994 +j0.06643
	p 2 = -0.7015 -j0.7127	0.92994 -j0.06643
Initial Zeros:		z1= 0.6
		z2 = 0.1
Initial Poles:	p1 = -1.1	0.9904
	p2 = -0.1	0.8957
Final Zeros:		z1 = 0.59787 error = 4.3%
		z2 = 1.01606 error = 3.5%
Final Poles:		p1 = 0.94188 + j 0.0674
		p2 = 0.94188 - j 0.0674
Pole Magnitude Error	1.2%	

After the gradient filters were properly initialized, the filter was allowed to adapt for 80K iterations. The LMS gain for the numerator coefficients was set to 0.03125 and the LMS gain for the denominator coefficients was set to 1e-6. Every 8K iterations the numerator and denominator coefficients were inspected and the and the locations of the poles and zeros of the discrete time model were calculated. Figures 33 - 38 show the behavior of the filter.

FIGURE 33. Output Error Experiment Numerator Coefficient Trajectories

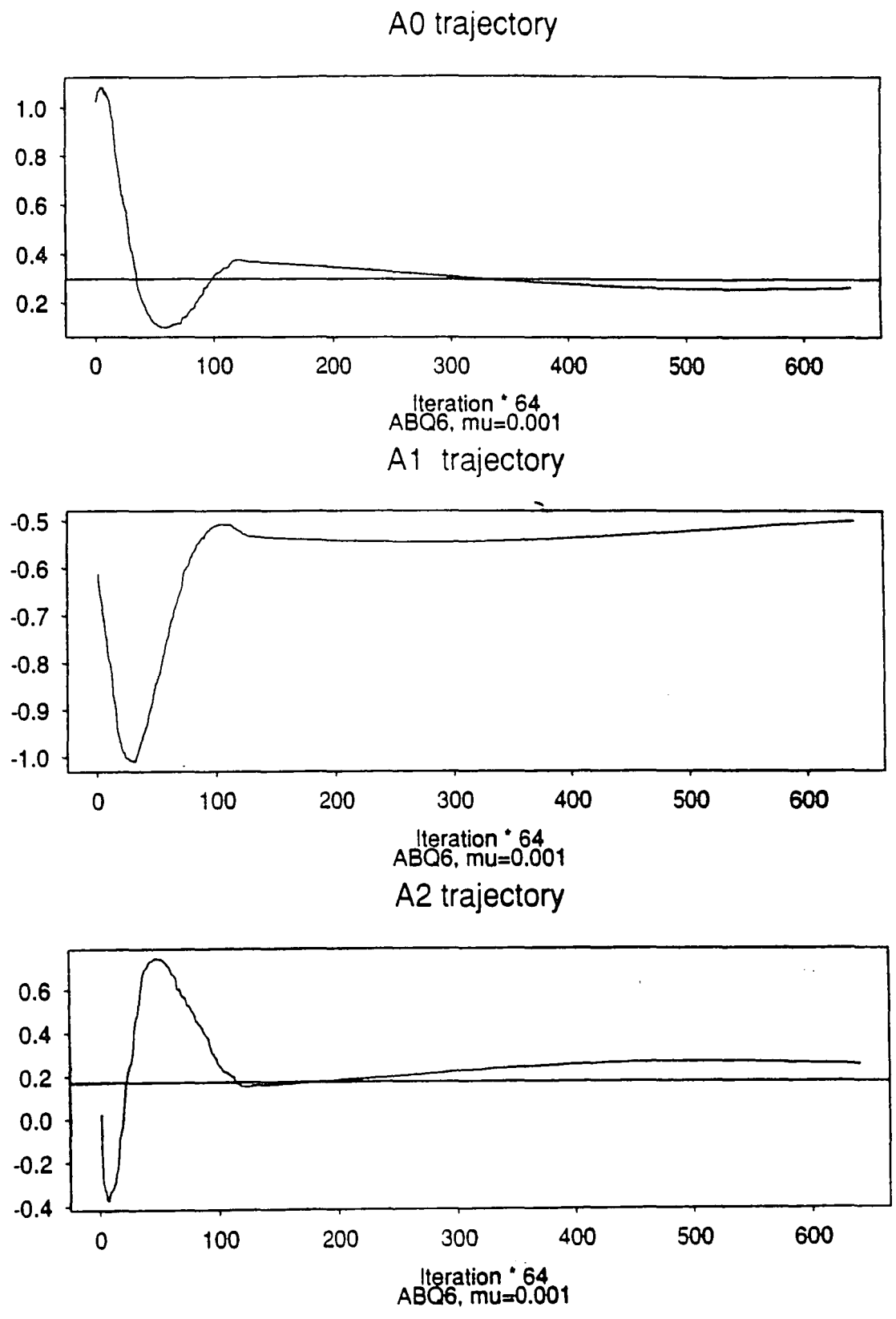


FIGURE 34. Output Error Formulation Denominator Coefficient Trajectories

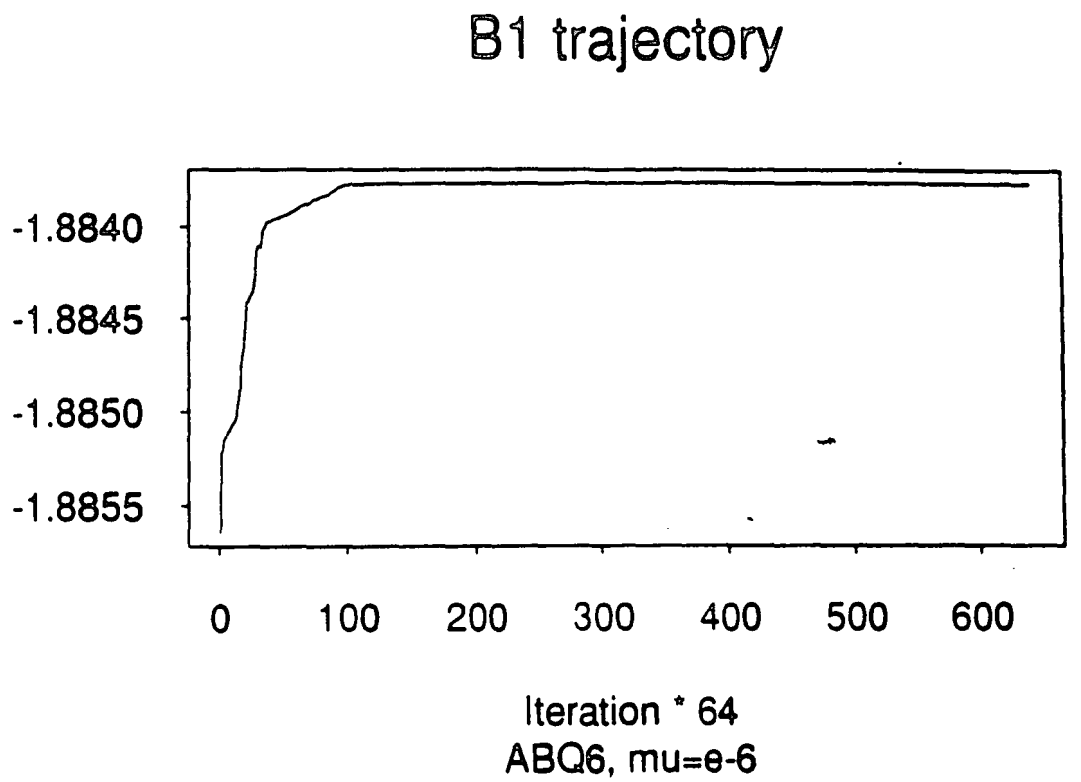


FIGURE 35. Output Error Experiment Pole Movement Over First 8K

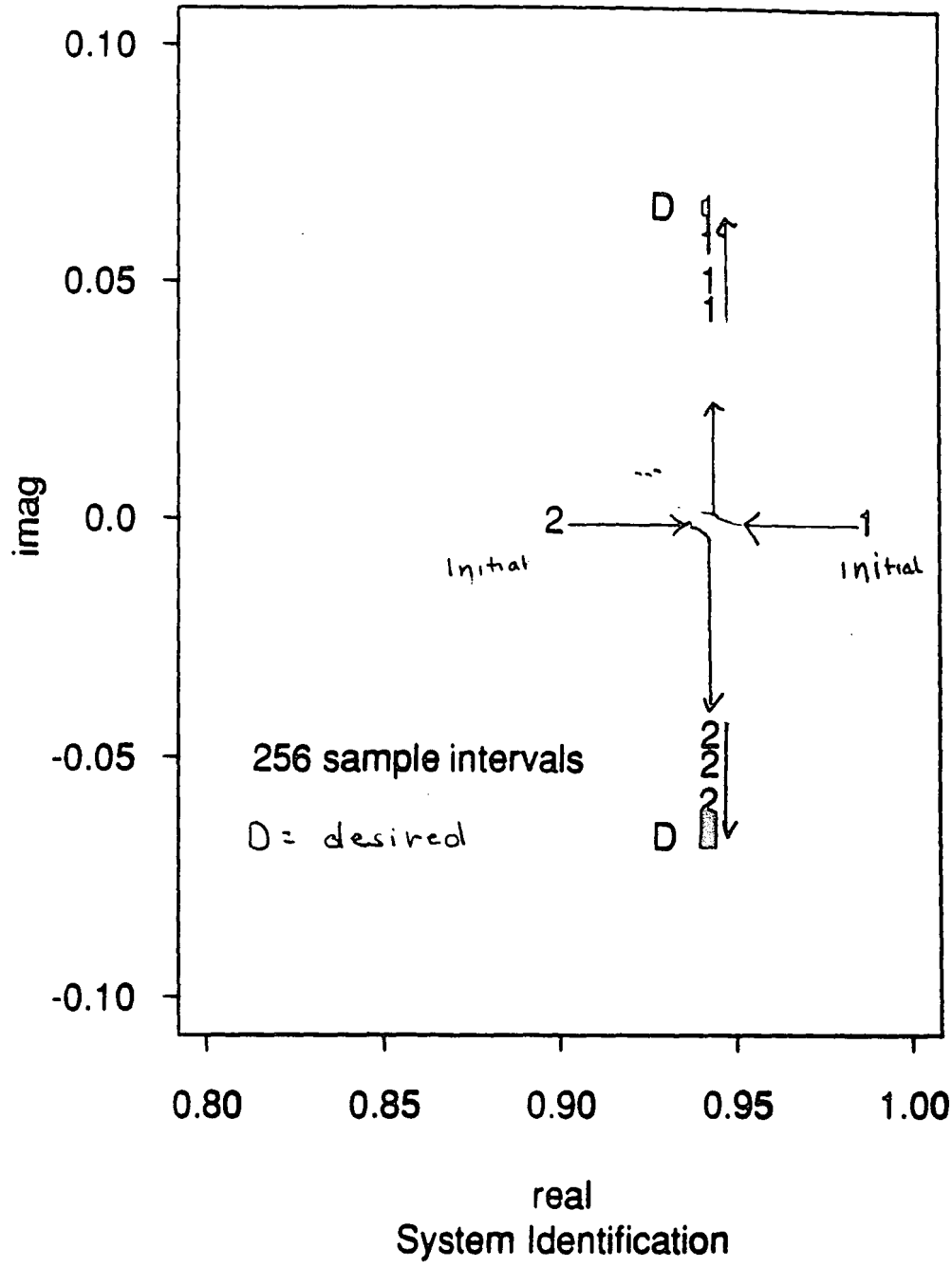


FIGURE 36. Output Error Experiment Zeros Movement Over 80K Iterations

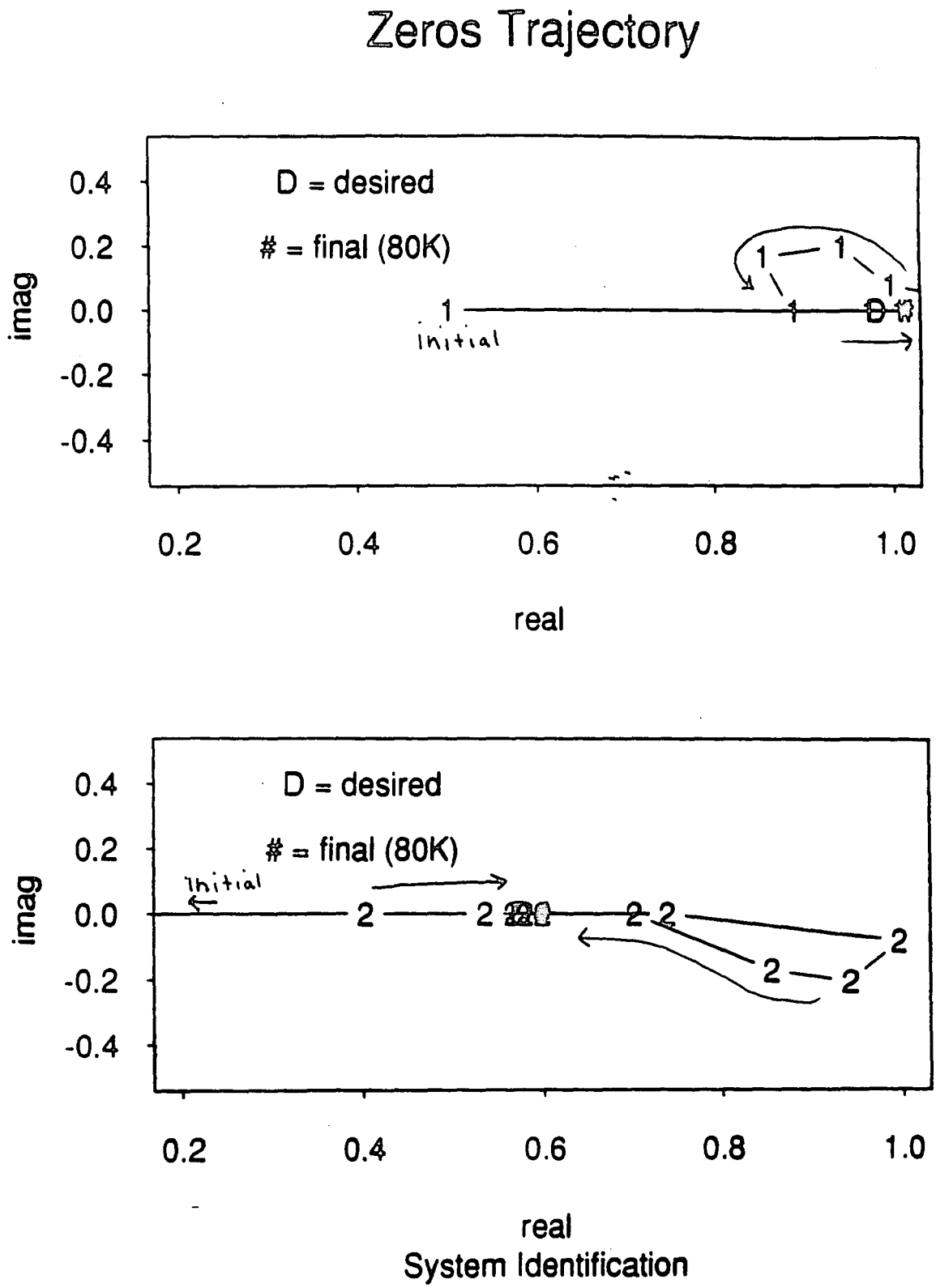


FIGURE 37. Output Error Experiment Learning Curve

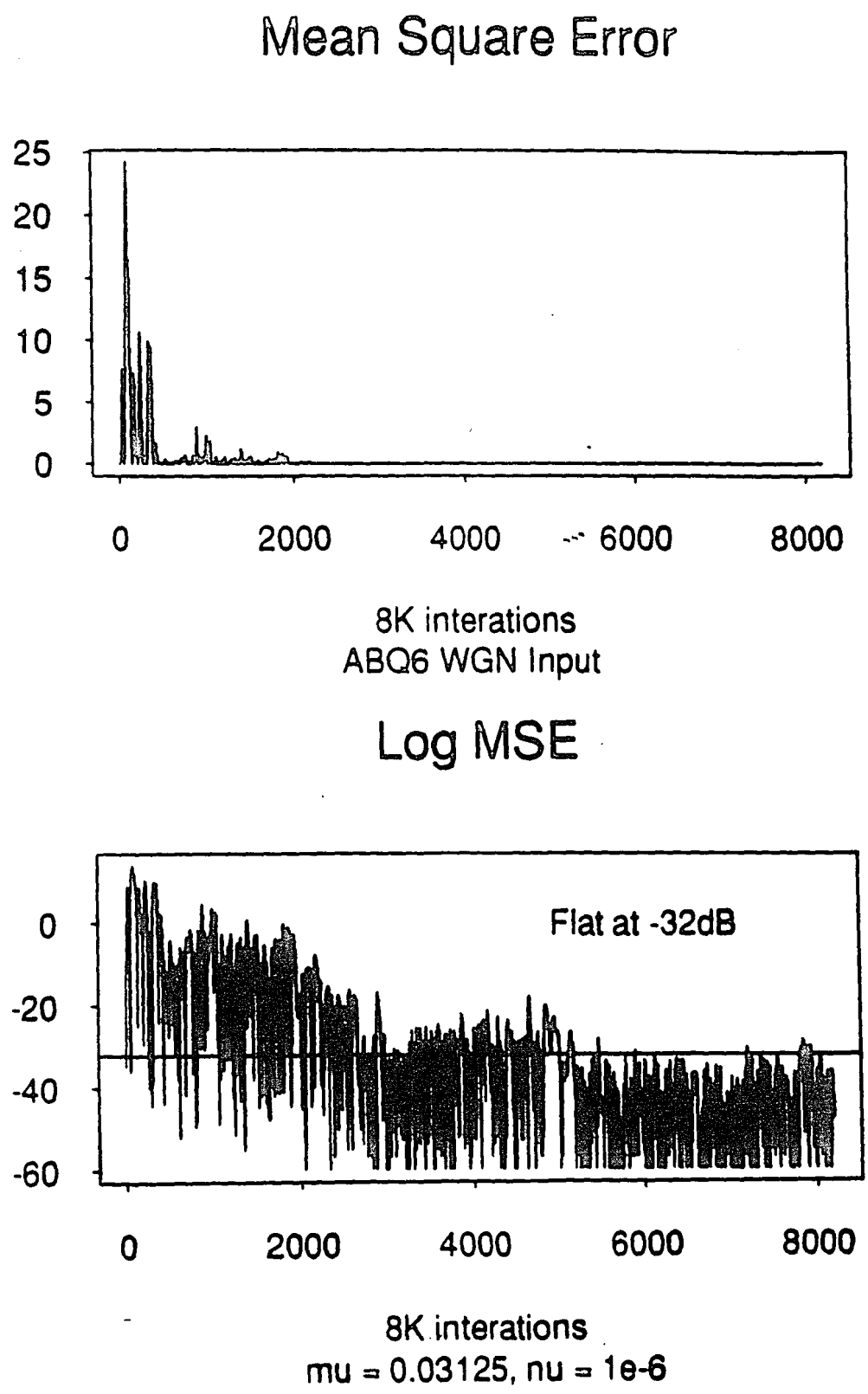


FIGURE 38. Output Error Formulation Transfer Functions

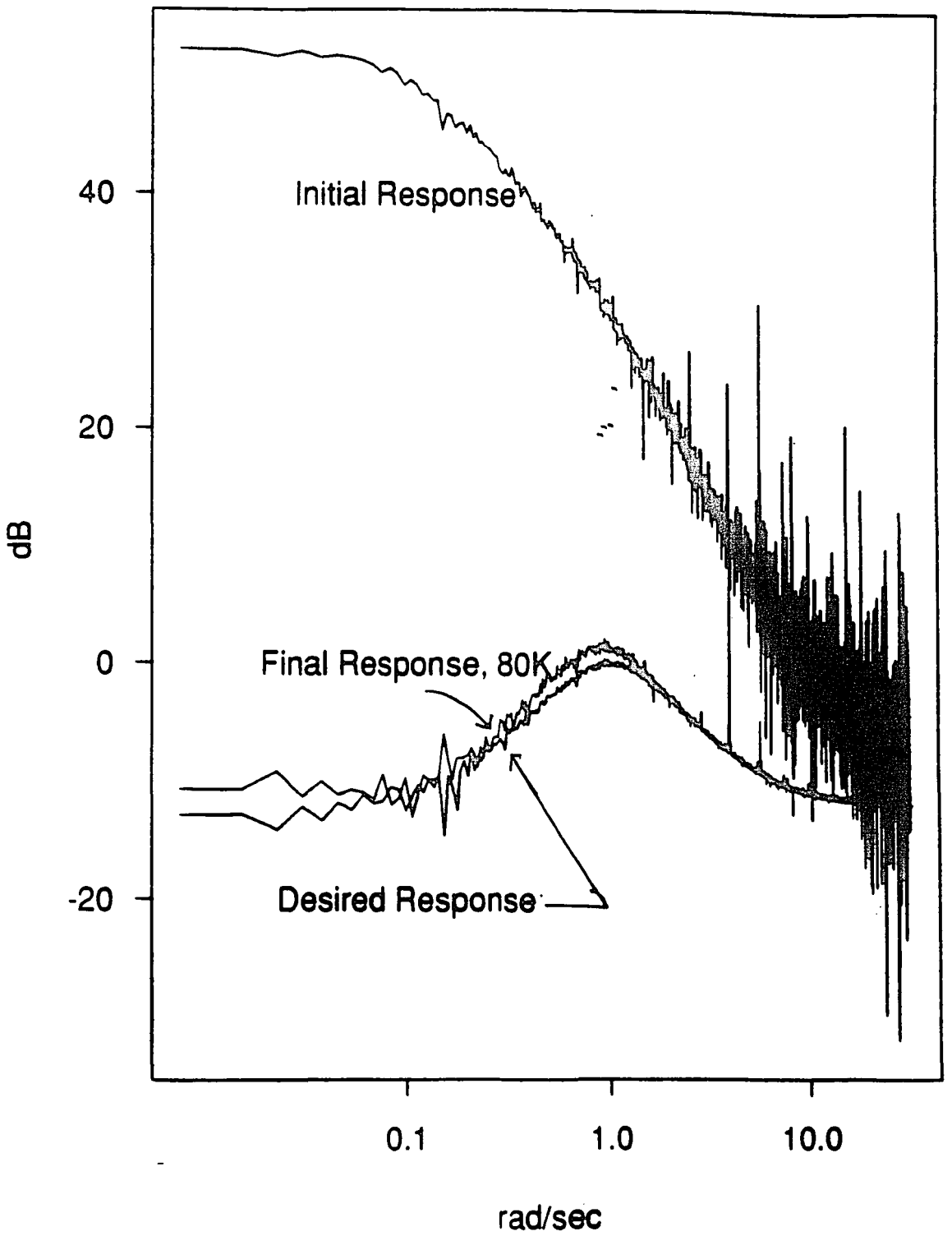


Figure 33 shows the trajectories of the numerator coefficients; Figure 34 shows the trajectories of one of the denominator coefficients. More illustrative of the behavior of the filter are Figures 35 showing the movement of the poles over the first 8K samples and Figure 36 showing the movement of the zeros over the 80K iterations

Referring to Figure 33 we see that the numerator coefficients vary fairly wildly over about the first 6400 iterations. After the initial transient period they begin slowly approaching the desired values. Figure 34 shows that the denominator coefficient also change very rapidly over the first 6400 cycles but then very slowly for the rest of the experiment. Figure 35 shows the story. The two poles were initialized as real poles that approximately straddled the location of the desired poles. The adaptation process changes the feedback of the system which causes the poles to move together then split and move off of the real axis. As the poles approach the desired poles they move more and more slowly; the final pole mis-adjustment is small. The system poles after 80K iterations have a magnitude of 0.944288 and an angle of 4.093 deg. The desired poles have a magnitude of 0.93231 and an angle of 4.086 deg. The error in the magnitude of the poles is less than 2%. After 80K the pole are still moving very slowly; it is not known how close they would eventually move. The final mis-adjustment will be limited by the adaptation gain of both the zeros and the poles.

Figure 36 shows the movement of the zeros. After passing the desired locations the zeros circled back around and have settled down very close to the location of the model zeros. Figures 37 and 38 show the learning curve and the transfer functions of the filter.

From these results we can conclude that the system is working and that we have a created a workable continuous time adaptive filter. Of course the problems of initialization and the stability of the filter and the algorithm have not gone away.

5.2.2 Initialization of The Filter Poles

The poles were not initialized where they were by accident. It was only after several attempts at making the system work and studying the movement of the poles that it was realized that the proper way to initial the poles is so that they tend to straddle the expected location of the desired poles. This implies that in a real application one would have to have a reasonable expectation of what the final system will look like. For many potential applications this is a reasonable requirement. What seems to be clear is that the poles behave as one expects the poles of a feedback system to behave under the influence of varying feedback. If real, they tend to move together along the real axis, then split and head towards the zeros. This implies that one may be able to use root-locus techniques to study the behavior of adaptive continuous time filters although this is beyond the scope of this thesis. Figure 39 shows the movement of the poles for an improperly unitized system. The desired poles are not bounded by the location of the initial poles. Under the influence of the adaptation control loop the poles move together and split into a complex conjugate pair. Because they leave the real axis on a trajectory far from the desired poles the system has no hope of converging to the true desired solution. However, in this process the filter still tends to minimize the mean square error. Figure 40 shows the learning curve, magnitude only, for this simulation. It appears that the filter has reached something like a local minimum. In addition note that in Figure 39 the poles are still moving fairly steadily on a track that will eventually take them outside of the unit circle. It was also observed that if the poles were initialized to far from the desired poles then it was very likely that the system would quickly drive the filter into instability. This was observed in a number of different cases. A final observation was that it appears that once the poles have become complex they appear to maintain the same radius. It is may be that addition of a third adaptive pole may help. There is some evidence to suggest that the order of the adaptive

filter should always be higher than the order of the system being identified [14]. In this thesis both systems were second order and very reasonable results were obtained.

In addition to poorly chosen initial poles, the prime cause of the filter becoming unstable was having the LMS gain for the denominator coefficients to large in comparison with the LMS gain for the numerator coefficients. It seems that the error is so sensitive to the location of the poles that they must be adapted with a much smaller gain than the zeros. Even in this case with ν two orders of magnitude smaller than $\mu$the poles move much faster than the zeros. Figures 35 and 36 clearly show this. It may turn out that this is a major problem in the application of CT adaptive filters.

5.2.3 Adaptation Of the Poles With Correlated Input Signals

In order to study the behavior of the adaptation of poles in a little more depth another experiment was conducted with fixed perfect zeros and adaptive poles.This was done with both WGN and with a multi-tone input signal. In both cases it was found that the filter adapted properly but much more slowly than before. The reason for the slower adaptation rate seems to be the much smaller error present when the zeros are already perfectly adaptive. This may be a topic for further investigation Figure 31 shows the movement of the poles for the filter with the 10-tone multitone input signal at 8K intervals. The poles are moving in the correct direction. The adaptation rate is excruciatingly slow. In 80K iterations the poles have not moved as far as they did in 8K iterations in the earlier experiment (Figure 35). The numerator gain was even increased an order of magnitude to 1e-5 for this simulation.

Using the multitone input signal a number of simulations were run with various signal-to-noise ratios. In the same manner as was done for the adaptive zeros/fixed poles filter the filter was initialized and allowed to adapt for 8K. The MSE was calculated and plotted every 1K iterations as shown in Figure 42. The adaptation process works very well in reducing the MSE down to 10dB S/N. At 0 and -6dB there is less error reduction. The

0dB curve exhibits a little divergence that is believed to be part of the rippling behavior of the MSE.

5.2.4 Summary

In this section simulation results for the output error adaptation of a continuous time filter were presented. The filter was found to be functional and to provide very good adaptation over a range of signal to noise ratios. It was shown how the poles should be initialized to obtain convergence and how best results were obtained with the numerator LMS gain two to four orders of magnitude less than the denominator LMS gain. Finally, it was suggested that root locus techniques may prove very useful in studying the behavior of CT adaptive filters. While much more difficult to use and requiring more hardware than the adaptive zero, fixed pole, filter, the output error continuous time adaptive filter may prove useful in highly specialized applications. One such application is automatic tuning of integrated continuous time filters.

FIGURE 39. Pole Movement for Improperly Initialized Poles

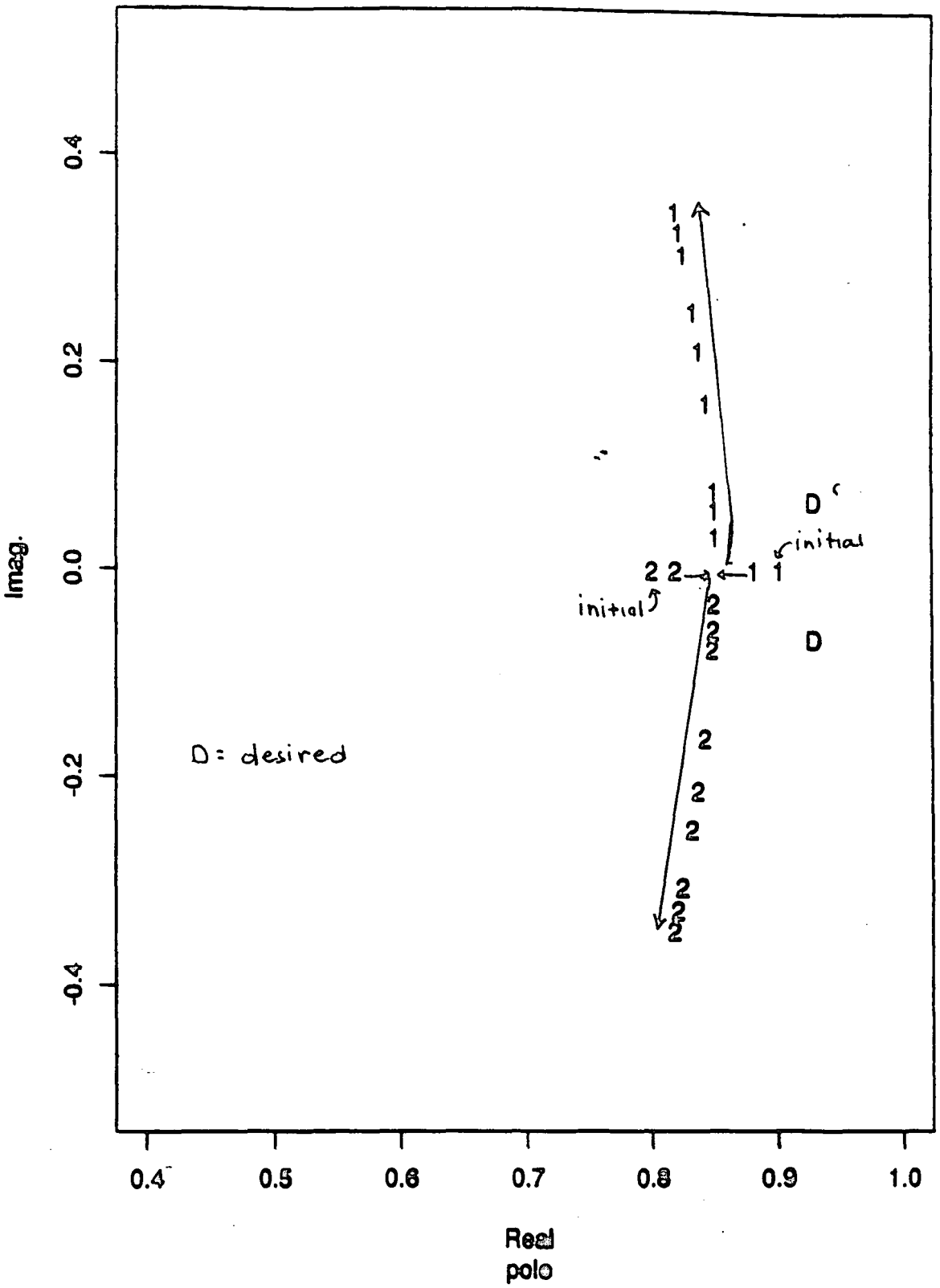


FIGURE 40. Learning Curve for Improperly Initialized Poles

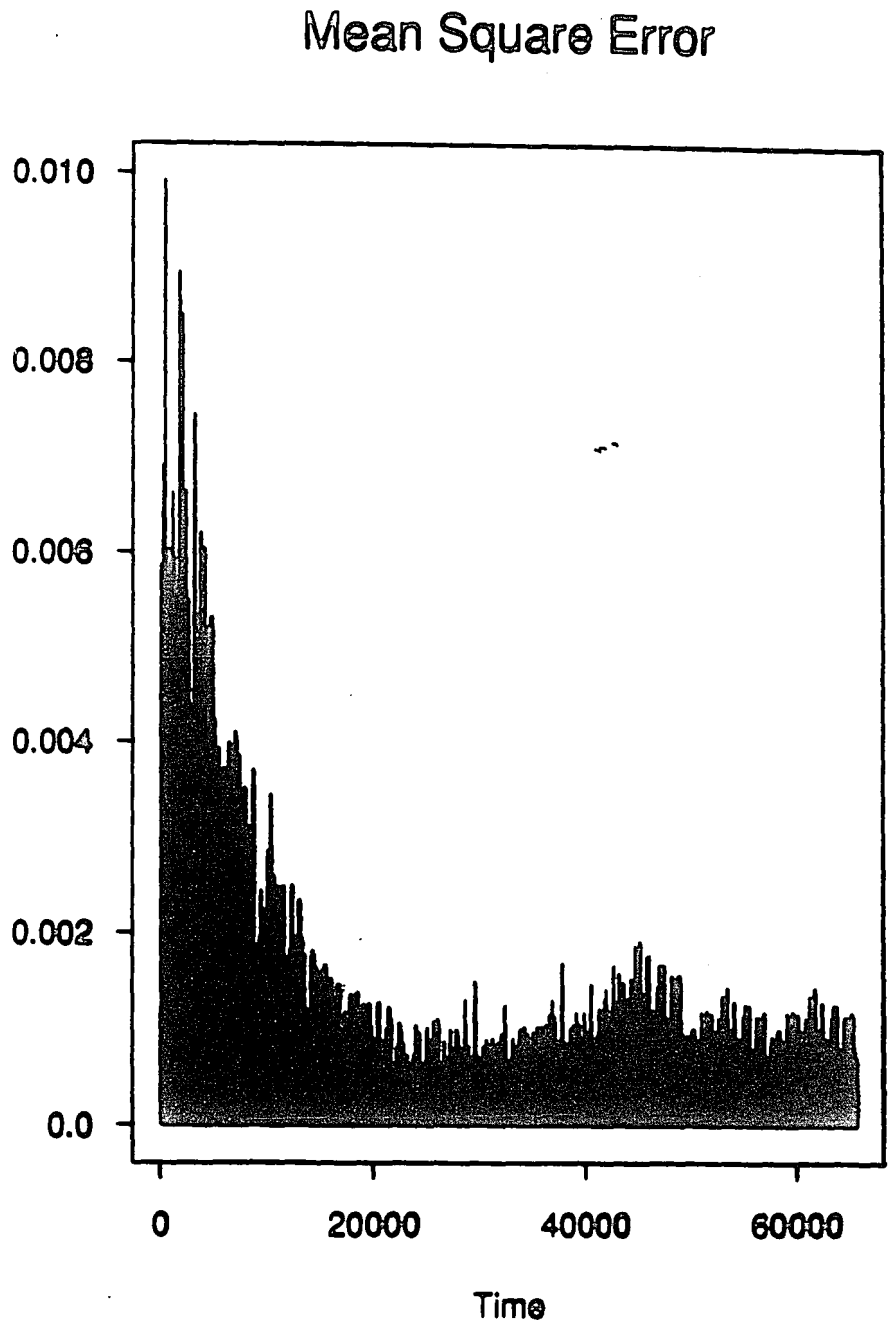


FIGURE 41. Pole Movement with Multi-tone Input

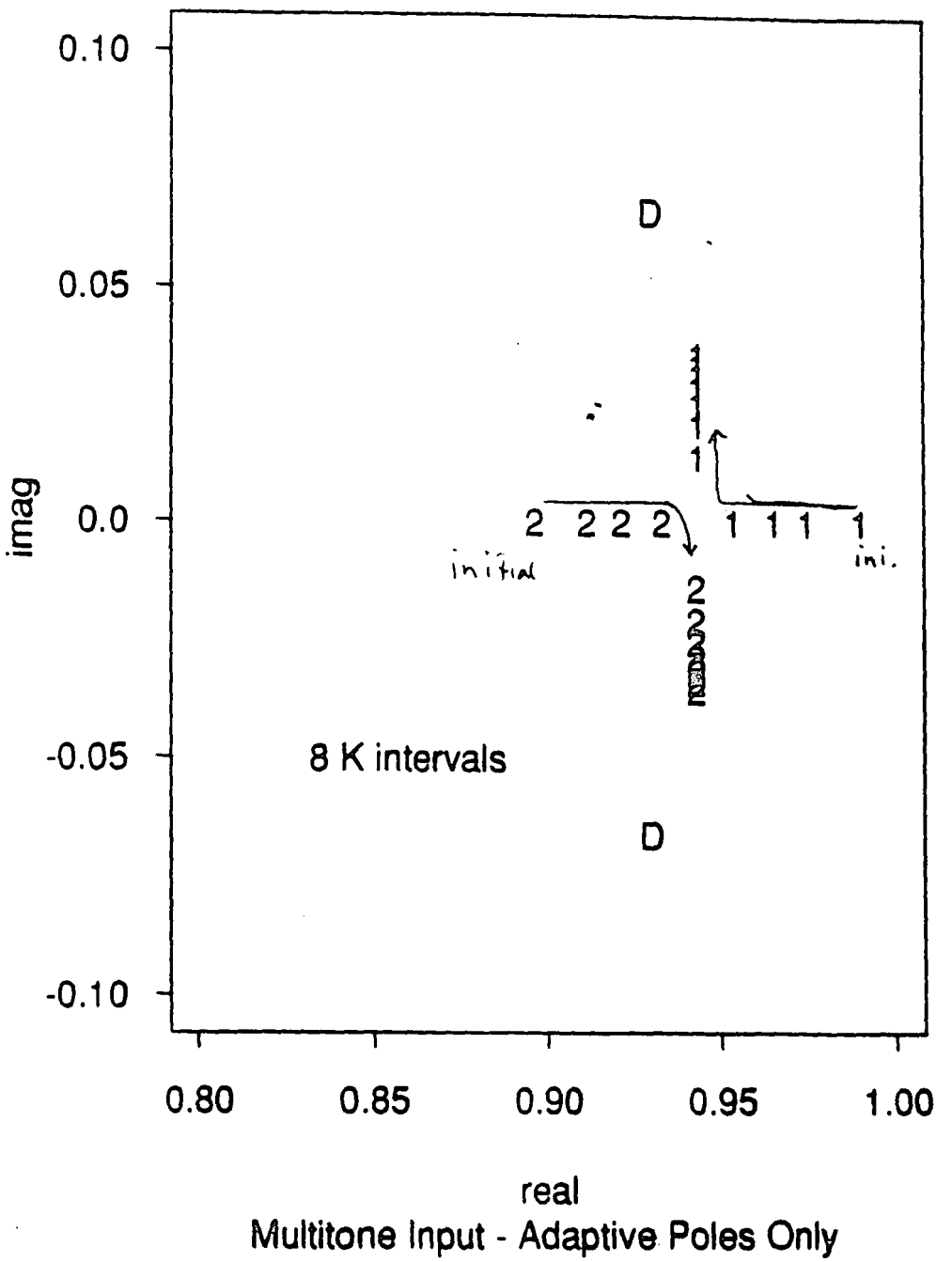
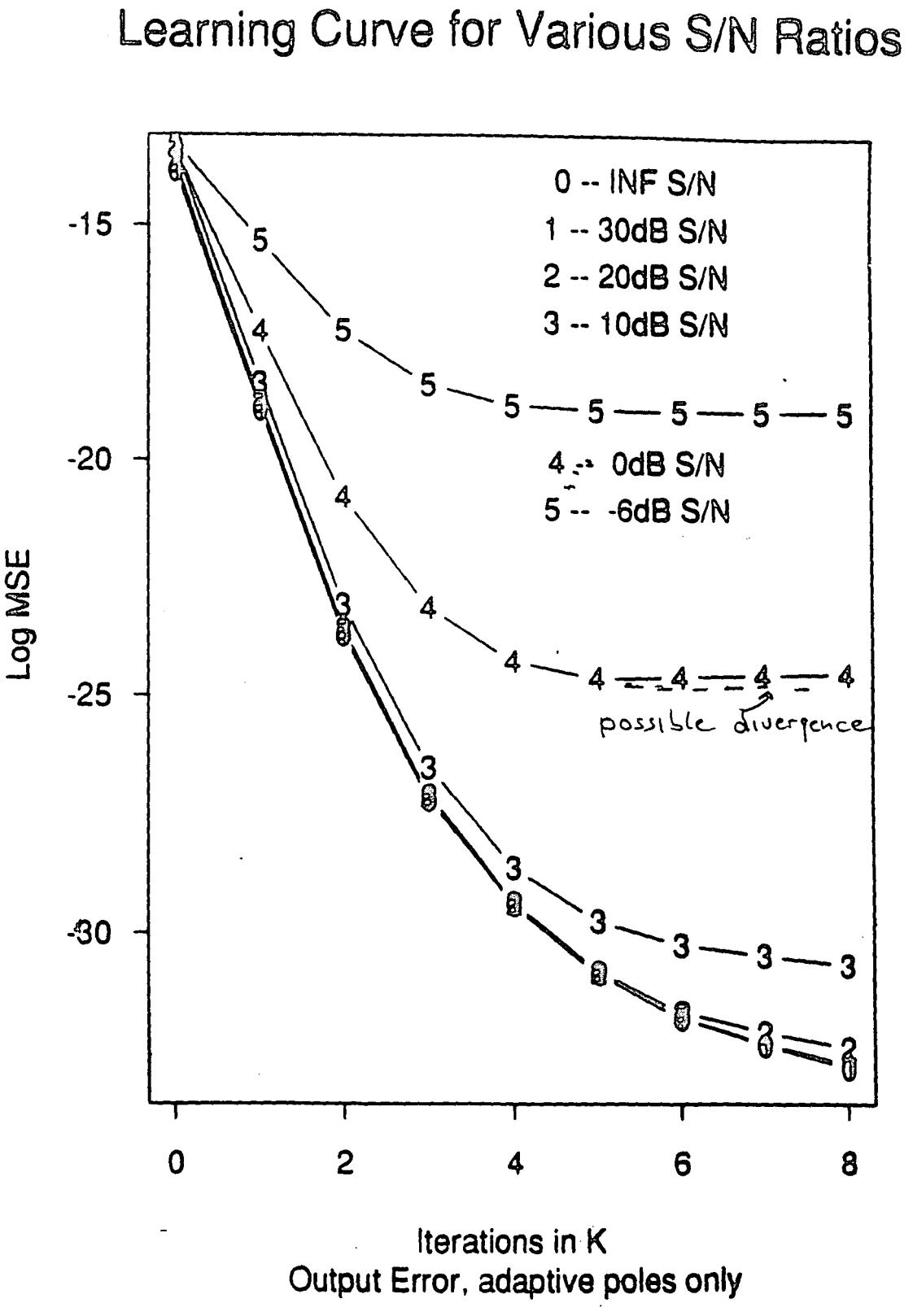


FIGURE 42. Learning Curves for Various S/N Ratios



5.3 Equation Error Adaptation Experiments

In order to test the equation error formulation of the continuous time adaptive filter, two simple simulations were conducted. Both were system identification problems with the amplitude equalizer as the unknown system. Gaussian noise was used as the input in both cases and the filters were allowed to adapt for at least 80K iterations. The poles and zeros were initialized exactly as they were for the output error experiments. In one case a butterworth polynomial with a cutoff frequency of 4 rad/s was used as the dummy denominator In the second experiment the same butterworth polynomial was used but the cutoff frequency was 1 rad/ sec. It seems somewhat intuitive that the dummy filter should be a low Q lowpass filter, but this has not been proven. The filter parameters are given in Table 11.

TABLE 11. Equation Error Experiments

	Continuous Time	Discrete Time
Zeros	-0.18432	0.9814
	-5.42262	0.5730
Poles:	-0.7015 +j0.7126	0.92994 +j0.06643
	-0.7015 -j0.7126	0.92994 -j0.06643
Dum. 1	-0.8298 +j4.820	0.92034 +j0.61159
	-0.8298 -j4.820	0.92034 -j0.61159
Dum 2	-0.7071 -j0.7071	0.92945 + j0.06588
	-0.7071 -j0.7071	0.92945 - j0.06588
Init		0.6
Zeros		0.1
Initial	-1.1	0.9904
Poles	-0.1	0.8957

For these simulations the LMS gains were set to 0.03125 for the numerator coefficients and 0.001 for the denominator coefficients. It was found that the gain for the denominator coefficients once again needed to be smaller than the gain for the numerator coefficients, but not as small as for the output error filter.

For these conditions it was found that both filters converged to the region of the desired response, and while they did not converge on the exact same locations, there is not

enough difference to draw conclusions on the effect of the dummy filter. In both cases there was mis-adjustment in both the magnitude and angles of the poles after 80K iterations. More disturbing is that it appears that the poles have not stopped moving. Figure 43 shows the pole movement for the first experiment; Figure 44 shows the pole movement for the second experiment. Table 12 shows the final pole and zero locations for both experiments.

TABLE 12. Equation Error Results

	Experiment 1	Error	Experiment 2	Error
Zero	0.98877	0.71%	1.0412	5.7
Zero	0.58083	1.35%	0.39239	46%
Poles	.94273 @ 6.74 deg	1.2%	.9458 @ 5.64 deg	1.4%

This table shows that while in both cases the poles converged to within 2% of the desired values and in the first case the zeros also converged to about the same error, in the second experiment the zeros suffer from significant mis-adjustment. This may or may not be related to the characteristics of the dummy denominator. A significant amount of work needs to be done to fully characterize and understand this system. The fact that there is still significant bias in the location of the poles, and that the bias is about the same in both the equation error and output error formulations, is also interesting and may indicate an unrecognized problem with the simulator programs. None of these are particularly easy questions to answer. Figures 45 and 46 show the movement of the zeros for the first and second experiments. Figures 47and 48 show the learning curves for both experiments. In both cases the error is still decreasing at the end of the allowed adaptation interval. The strong "blips" in the learning curve shown in Figure 47 are the result of restarting the simulation process and not true spikes in the MSE.

FIGURE 43. Pole Movement for Equation Error Experiment #1

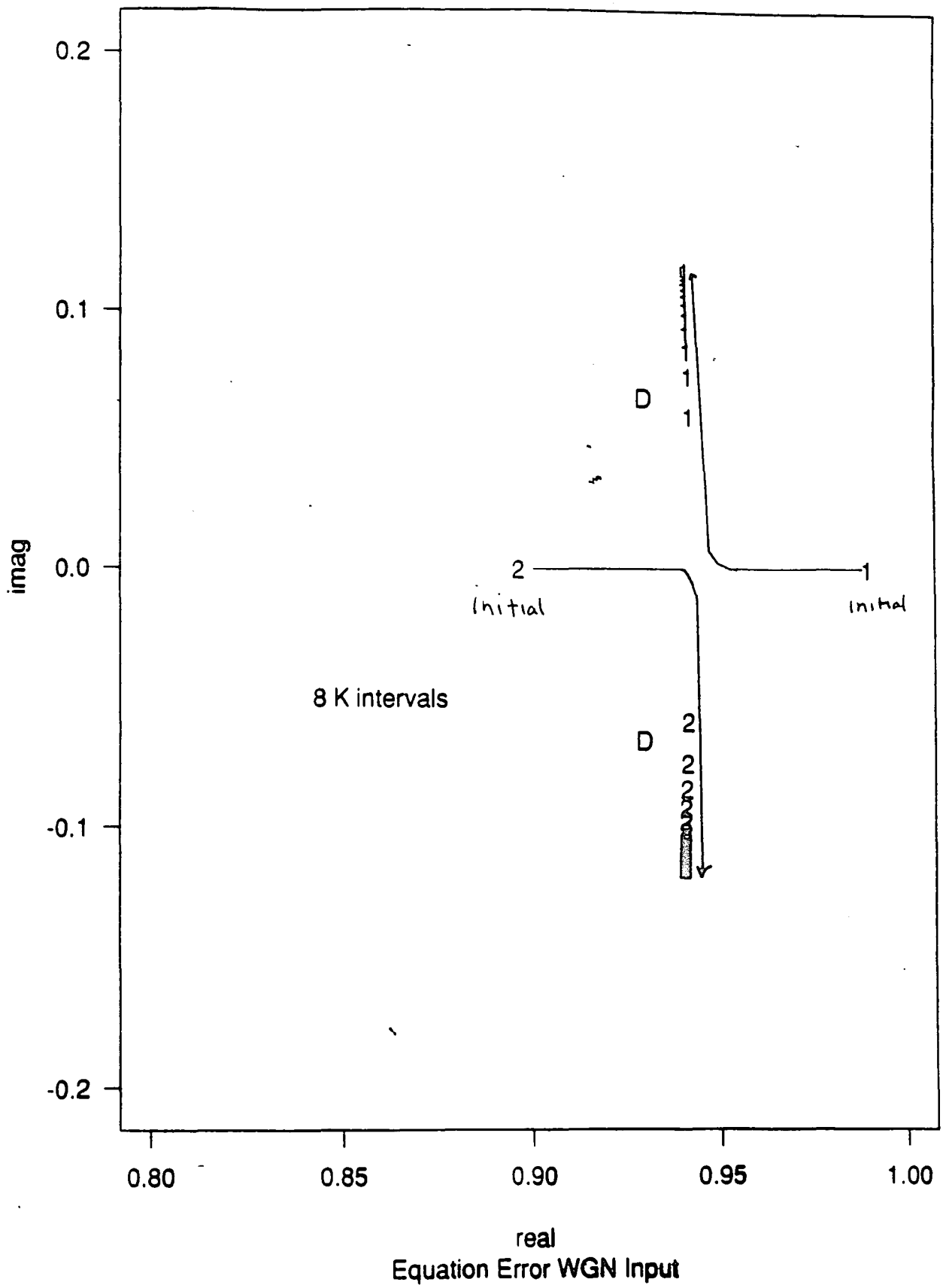


FIGURE 44 Pole Movement for Equation Error Experiment #2

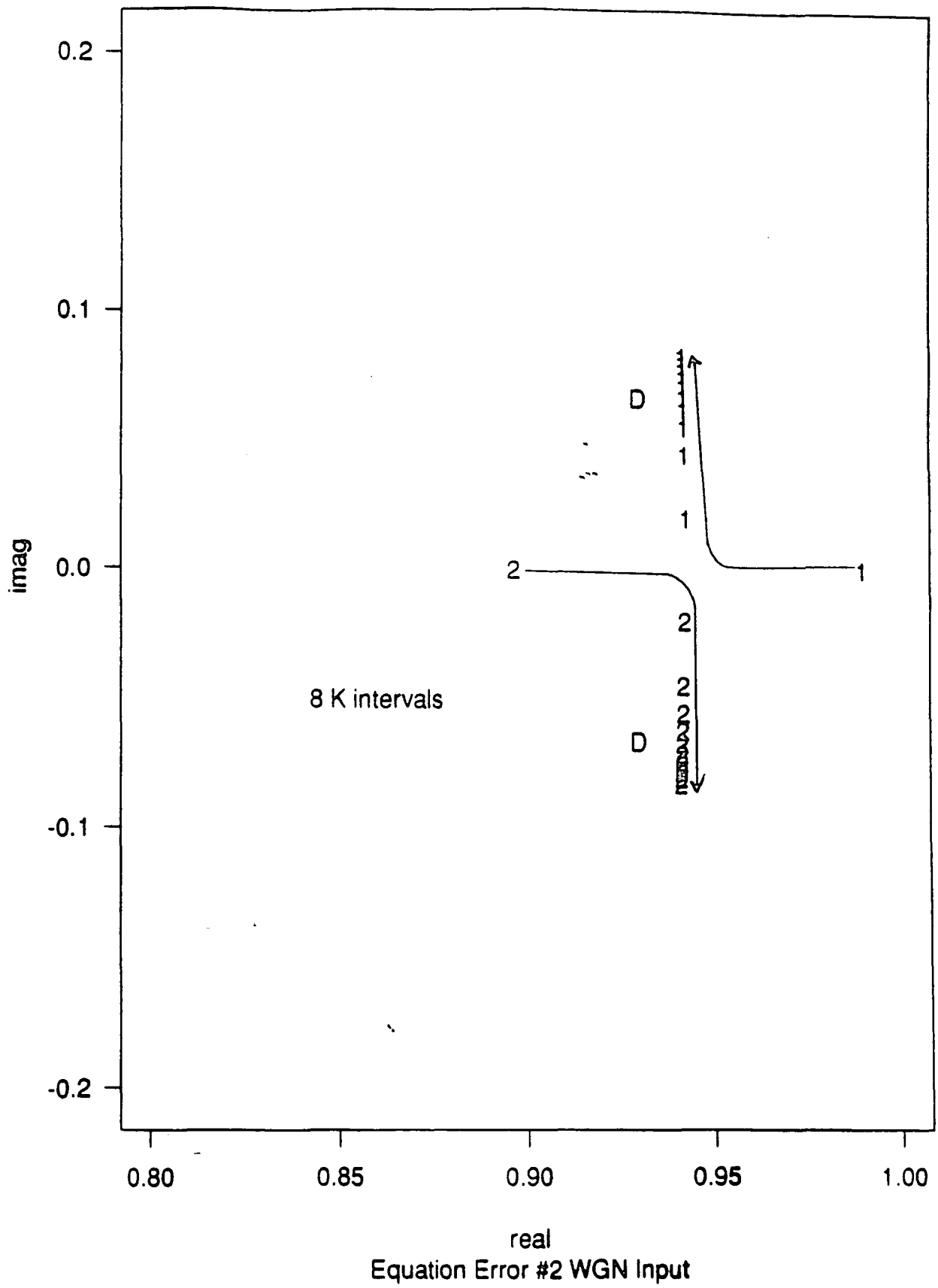
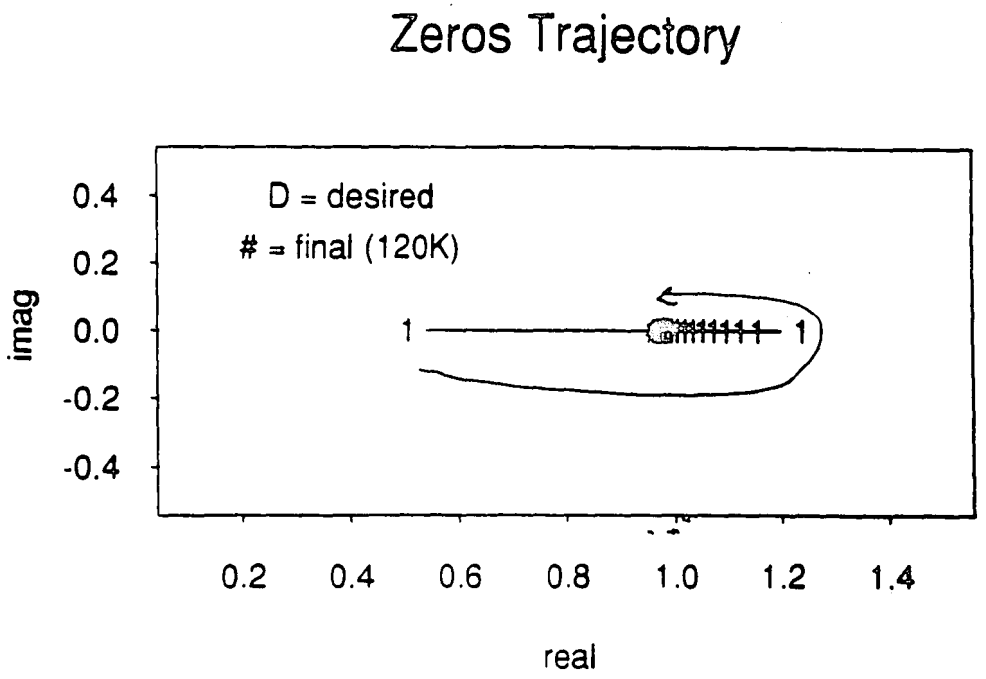
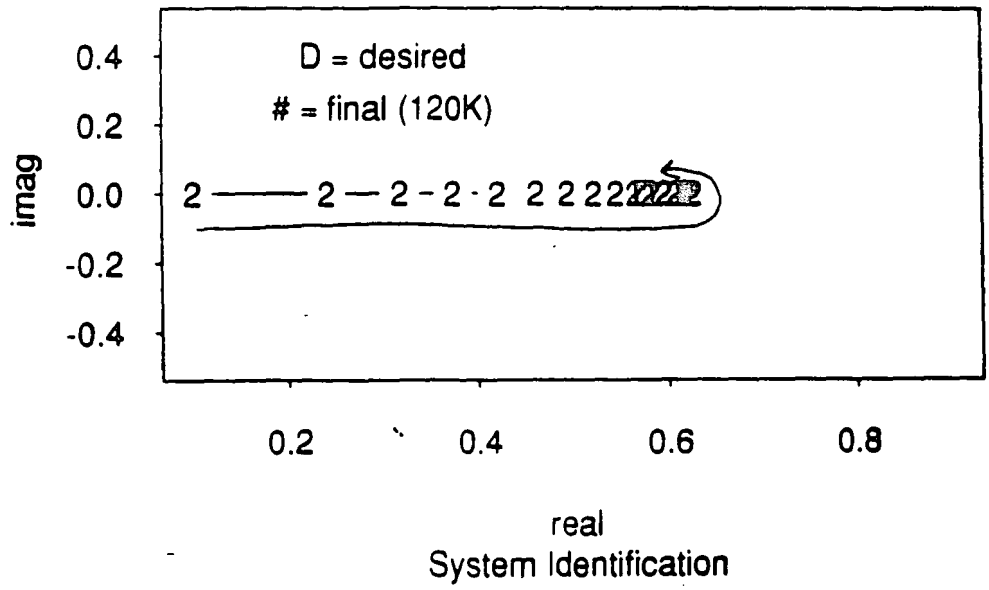


FIGURE 45. Zero Movement for Equation Error Experiment #1





System Identification

FIGURE 44. Zeros Movement for Equation Error Experiment #2

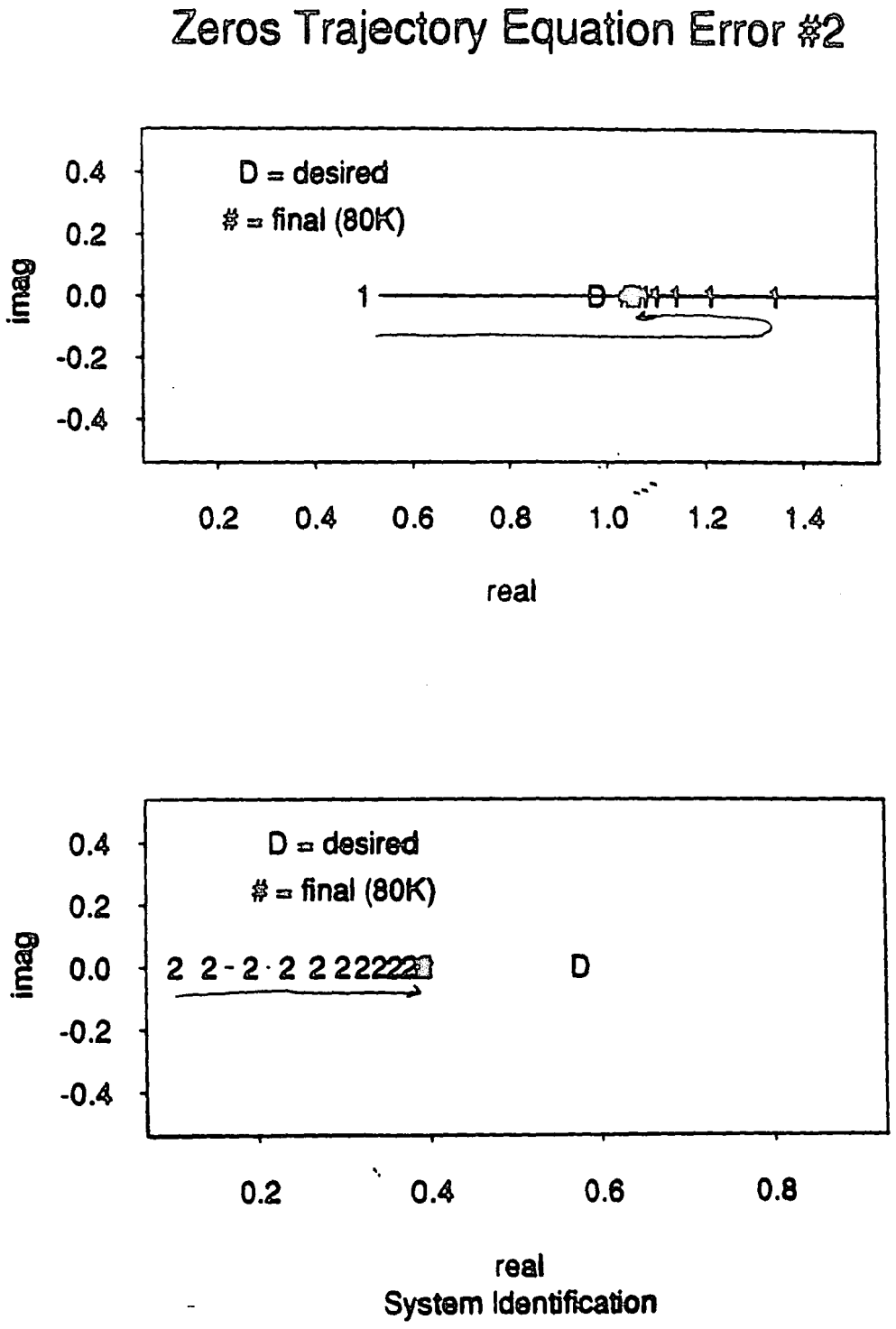


FIGURE 47. Learning Curve for Equation Error Experiment #1

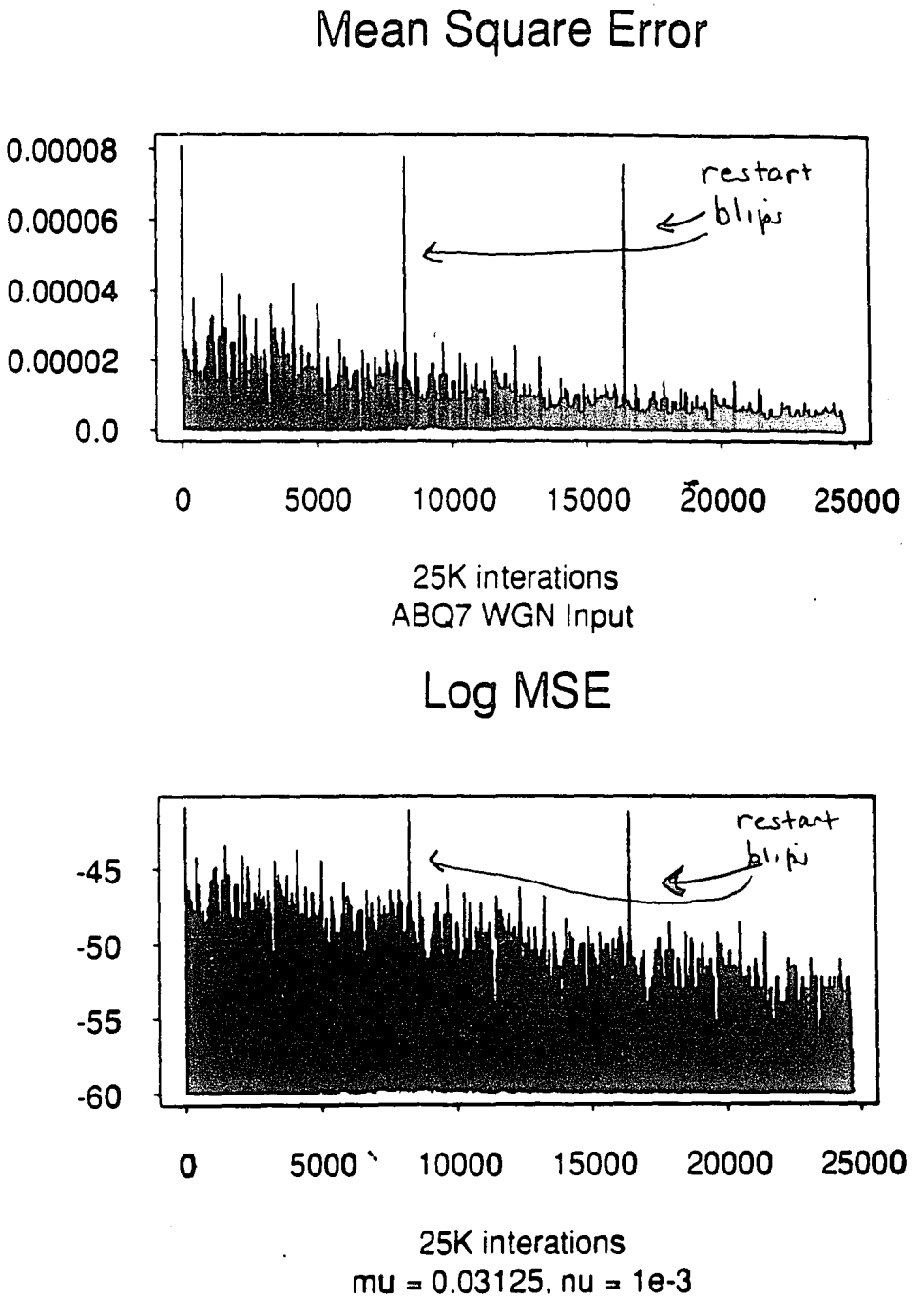
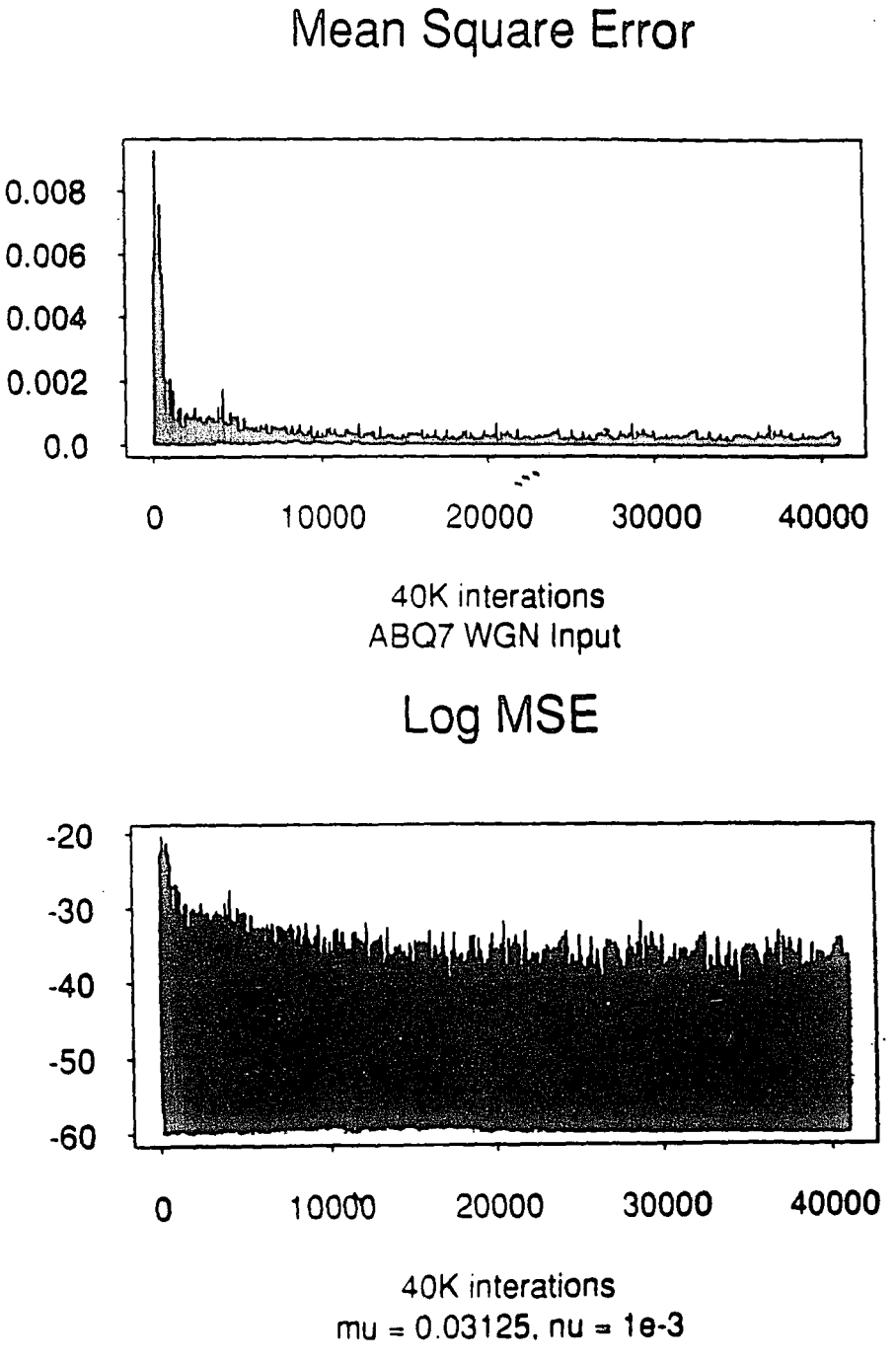


FIGURE 48. Learning Curve for Equation Error Experiment #2



5.3.1 Summary

One of the interesting observations made during these simulations is that it was found to be necessary to initialize the poles in exactly the same manner as it was for the output error formulation. Even though we are adapting the zeros of each of the equation error filters the filter that is attempting to identify the poles moves the poles along the same sort of loci as the output error filter. It does so, however, at a slower rate. This may be useful in that it would seem that the slower adaptation rate should allow the system to adapt with less coefficient bias than before. This was not found to be true in the limited number of simulations done for this thesis.

What is significant is that the equation error formulation does work. While this implementation of an adaptive continuous time filter requires more hardware, and obviously more study, it does offer a solution to the problems associated with adapting the poles of a continuous time filter.

6.0 Conclusions

This thesis has presented and discussed three methods of building continuous time adaptive filters. Equations have been derived showing how these filters can be adapted in real time by using the LMS algorithm. There is no requirement for large amounts of complicated digital hardware or signal processing. All of the functions required can be readily implemented using standard analog design techniques.

The first and simplest filter proposed here is a continuous time biquadratic filter with adaptive zeros and fixed poles. Except for a few analog multiplier, this filter requires very little extra hardware beyond the regular biquadratic filter. Many potential applications for analog adaptive filters can be served by a filter with only adaptive zeros. Simulations showed that this filter was robust and converged with a wide variety of input signals. It was found that the LMS gain along with the location of the eigenvalues of the input auto correlation matrix controlled the stability and rate of adaptation just as they do for the standard FIR filter implementation. The error impulse response was found to be useful in studying the stability and adaptation rate of the filter. This filter was extensively simulated and its application as an interference canceller was demonstrated.

The second filter proposed is a biquadratic filter with adaptable poles and zeros based on the output error formulation of a recursive adaptive filter. While more complicated and difficult to use, simulations have shown that this filter can be made to work in a systems identification application if properly initialized. It was found that the poles of the filter needed to be initialized on the real axis such that they straddled the locus of the desired poles. This implies that standard root locus techniques may be useful in studying this system. While probably less useful than the additive zero, fixed pole filter, this filter may be useful in specialized applications including automatic filter tuning.

The third system proposed uses the first type of filter to build what is known as the equation error formulation of a recursive adaptive filter. While this requires more

hardware than the other two methods it does surmount some of the problems encountered when trying to adapt the systems poles. This system is well known for digital filters but it does not seem to have ever been tried for a continuous time filter before. It was found that this filter required the same type of initialization of the poles as the output error filter. It was also found that this filter suffered from more bias than the output error filter. It is suspected that this bias is related to the behavior of the dummy filter and much more work will be required to truly understand this system. While this last system was not extensively studied for this thesis, it was simulated and found to work.

It was not possible for this work to delve into any one of these filters to the depth that is really required. Each system could well be the topic of yet another thesis. A great deal of work needs be done to properly analyze these systems. Some specific topics that should be addressed are a detailed analytical study of the effect of the feedback in the filter on the LMS algorithm. It would be very useful if it could be learned how to apply root locus techniques to the study of these filters. It would also be worthwhile to explore the error surfaces of any of these filters to try and understand how to prevent local minimums and instability problems. Also it would be extremely educational to actually develop circuits to implement these structures and experiment with them in the laboratory. While the simulations done in this work have shown the feasibility of these systems and have uncovered some of the problems and pitfalls associated with them, only real circuit experience can turn adaptive continuous time filters from potential systems to real solutions.

[1].Baher, H, Analog and Digital Signal Processing, J. Wiley & Sons Inc., New York, 1990.

[2].Bellanger, M.G.,Adaptive Digital Filters and Signal Analysis,Marcel Dekker Inc., New York, 1987.

[3].Bode, H.W, Network Analysis and Feedback Amplifier Design, Van Nostrand Press, New York, 1954.

[4].Brooks, T.L, "Simultaneous Tuning and Signal Processing in Integrated Continuous-Time Filters: the Correlated Tuning Loop,"Proceedings ISCAS, 1989, p. 651.

[5].Brown, R. G, Introduction to Random Signal Analysis and Kalman Filtering, John Wiley & Sons, Inc., New York, 1983.

[6].Chambers, J, Graphical Methods for Data Analysis, Wadsworth International Press, Belmont, CA, 1983.

[7].Cowan, C.F.N, Adaptive Filters, Prentice Hall, Englewood Cliffs, New Jersey, 1983.

[8].Fernandez, F.J, Private Communications, 1992.

[9].Feintuch, P.L,."An Adaptive Recursive LMS Filter," Proceedings of the IEEE, November, 1976, p. 1622.

[10].Haykin, S, Adaptive Filter Theory, 2d ed. Prentice Hall, Englewood Cliffs, New Jersey, 1991.

[11].Hsia,.T.C, "A Simplified Adaptive Recursive Filter Design," Proceedings of the IEEE, Vol. 69, No. 9, 1981, p.1153.

[12].Jeruchim, M.C, Simulation of Communications Systems, Plenum Press.,New York, 1991.

[13].Johns, D.A,W.M. Snelgrove and A.S. Sedra,"Adaptive Recursive State Space Filters Using a Gradient Based Algorithm," IEEE Transactions on Circuits and Systems, Vol. 37, No. 6, 1990, p. 673.

[14].Johns, D.A, W.M.Snelgrove and A.S. Sedra,"Continuous Time LMS Adaptive Recursive Filters," IEEE Transactions on Circuits and Systems. Vol 38, No. 7, 1991, p.769.

[15].Jenkins, W.K. and M. Nayeri,"Adaptive Filters Realized with Second Order Sections," Proceedings of the ICASSP, 1986, p 2103.

[16].Kozma, K, D.A. Johns and A.S. Sedra,"Automatic Tuning of Continuous Time Integrated Filters Using an Adaptive Filter Technique," IEEE Transactions on Circuits and Systems, Vol 38 No. 11, 1991, p. 1241.

[17].Kwan,T. and K. Martin,"An Adaptive Analog Continuous-Time CMOS Biquadratic Filter," IEEE Journal of Solid State Circuits" Vol. 26, No. 6, 1991, p 859.

[18].Kwan,T. and K. Martin,"Adaptive Detection and Enhancement of Multiple Sinusoids Using a Cascade IIR Filter," IEEE Transactions on Circuits and Systems, Vol. 36, No. 7, 1989, p. 937.

[19].Lucky, R.W, "Automatic Equalization for Digital Communication," Bell System Technical Journal, April 1965, p.547.

[20].Martin, K. and A.S. Sedra,"Switched-Capacitor Building Blocks for Adaptive Systems," IEEE Transactions on Circuits and Systems, Vol. CAS-28, No. 6, 1981, p 576.

[21].Martin, K. and M.T. Sun,"Adaptive Filters Suitable for Real Time Spectral Analysis," IEEE Journal of Solid State Circuits, Vol. SC-21, No. 1, 1986, p. 108.

[22].Mastrocola, A.R, Private Communications, 1992

[23].Mazo, J.E, "Analysis of Decision-Directed Equalizer Convergence," Bell System Technical Journal, December 1980, p.1857.

[24].Murano, K, "Echo Cancellation and Applications," IEEE Communications Magazine, January 1990, p 49.

[25].Parikh, D, "On an Adaptive Algorithm for IIR Filters," Proceedings of the IEEE, Vol. 66, No. 5, 1978, p. 585.

[26].Qiuting, H, "Offset Compensation Scheme For Analog LMS Adaptive FIR Filters," Electronic Letters, Vol. 28,No. 13.,1991, p. 1203.

[27].Qureshi, S.U.H, "Adaptive Equalization," Proceedings of the IEEE, Vol. 73, No. 9, 1985, p. 1349.

[28].Rabiner, L.R and B. Gold, Theory and Application of Digital Signal Processing, Prentice Hall, Englewood Cliffs, New Jersey,1975.

[29].Shynk, J, "Adaptive IIR Filtering," IEEE ASSP Magazine, April 1989, p. 4.

[30].Stephenson, F.W, RC Active Filter Design Handbook, John Wiley and Sons, Inc., New York. 1985.

[31].Stearns, S.D, "Error Surfaces of Recursive Adaptive Filters," IEEE Transactions on Circuits and Systems, Vol. CAS-28, No. 6, 1981, p.603.

[32].Treichler, J, "Transient and Convergent Behavior of the Adaptive Line Enhancer," IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-27, No. 1, 1979, p. 53.

[33].Voorman, J.O, et.al,"An Automatic Equalizer for Echo Reduction in Teletext on a Single Chip," Philips Technical Review, Vol. 40, November 1982, p.329.

[34].Widrow, B, Adaptive Signal Processing, Prentice Hall, Englewood Cliffs, New Jersey, 1985.

[35].Widrow, B. et. al, "Adaptive Noise Cancelling," Proceedings of the IEEE, Vol 63, No. 12, 1975, p 1692.

[36].Widrow, B. et.al, "Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter," Proceedings of the IEEE, Vol. 64, No. 8, 1976, p. 1151.

[37].Williams, C. S, Designing Digital Filters, Prentice Hall, Englewood Cliffs, New Jersey, 1986.

[38].White, M. et. al, "CCD Adaptive Discrete Analog Signal Processing," IEEE Journal of Solid State Circuits, Vol Sc-14, No. 1, 1979, p. 132.

[39].White, S, "Adaptive Recursive Digital Filter," Proceedings of the 9th Asilomar Conference on Circuits, Systems, and Computers, Nov. 1975, p. 21.

Vita

James M. Little was born on June 8, 1956 in Elgin Illinois. He is the son of Dorothy O. Little of Pottstown, Pennsylvania. He graduated from Pottstown High School in 1974 and received a Bachelor of Arts in the General ARts and Sciences from the Pennsylvania State University in 1978 and an Associate in Engineering in Electrical Engineering Technology from the Pennsylvania State University in 1983. From 1986 to 1989 he studied electrical engineering in evening classes at Penn State. In 1989 he enrolled at Lehigh University and is expecting to receive a Master of Science in Electrical Engineering in 1993.

Since 1983 Mr. Little has been employed by AT&T Bell Laboratories ia a variety of capacities. Since 1990 he has been designing custom mixed signal integrated circuits for high speed data communications systems.

END

OF

TITLE